

F²MC-8L/8FX FAMILY

SOFTUNE™ Workbench

OPERATION MANUAL

F²MC-8L/8FX FAMILY

SOFTUNE™ Workbench

OPERATION MANUAL

FUJITSU LIMITED

PREFACE

■ What is the SOFTUNE Workbench?

SOFTUNE Workbench is support software for developing programs for the F²MC-8L/8FX families of microprocessors / microcontrollers.

It is a combination of a development manager, simulator debugger, emulator debugger, monitor debugger, and an integrated development environment for efficient development.

Note: F²MC is the abbreviation of FUJITSU Flexible Microcontroller.

■ Purpose of this manual and target readers

This manual explains how to operate the SOFTUNE Workbench and design the product.

This manual is intended for engineers designing several kinds of products using SOFTUNE Workbench.

Be sure to read this manual completely.

■ Trademarks

SOFTUNE is a trademark of FUJITSU LIMITED.

Microsoft and Windows are registered trademarks in the United States and other countries of United States Microsoft Corporation.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Organization of Manual

This manual consists of five chapters and appendix.

CHAPTER 1 OUTLINE OF SOFTUNE Workbench

This chapter gives an outline of SOFTUNE Workbench.

CHAPTER 2 OPERATION

This chapter describes the basic operation method and development procedure of SOFTUNE Workbench.

CHAPTER 3 Windows

This chapter explains SOFTUNE Workbench windows.

CHAPTER 4 MENUS

This chapter explains the SOFTUNE Workbench menu configuration and the dialogs to be started from each menu.

CHAPTER 5 Add-in Module

This chapter explains SOFTUNE Workbench Add-in module.

APPENDIX

The Appendix describes the register name, downloading monitor program, setting LAN interface, setting USB interface, creating ROM on monitor debugger target, external I/F for simulator.

- The contents of this document are subject to change without notice.
Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of Fujitsu semiconductor device; Fujitsu does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. Fujitsu assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of Fujitsu or any third party or does Fujitsu warrant non-infringement of any third-party's intellectual property right or other right by using such information. Fujitsu assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

READING THIS MANUAL

■ Configuration of Page

In each section of this manual, the summary about the section is described certainly, so you can grasp an outline of this manual if only you read these summaries.

And the title of upper section is described in lower section, so you can grasp the position where you are reading now.

■ Product name abbreviation

In this manual and this product, product names are abbreviated as follows:

Microsoft® Windows® 2000 Professional operating system is abbreviated to Windows 2000.

Microsoft® Windows® XP Professional operating system is abbreviated to Windows XP.

CONTENTS

CHAPTER 1	OUTLINE OF SOFTUNE Workbench	1
1.1	Outline	2
1.2	What is SOFTUNE Workbench?	3
1.3	Procedure for Developing Programs with SOFTUNE Workbench	4
CHAPTER 2	OPERATION	5
2.1	Parameters to be Entered from Dialog	6
2.1.1	Data and Address Formulas (Numerical Constant)	7
2.1.2	Data and Address Formulas (Symbols, Line Numbers, Character Constants)	9
2.1.3	Data and Address Formulas (Register Name, Flag Name)	11
2.1.4	Operators Usable in Data and Address Formulas	12
2.1.5	Address Formula Specification	14
2.1.6	File Name Specification	15
2.2	Starting and Terminating SOFTUNE Workbench	16
2.3	Creating Workspace	17
2.4	Storing of Project	19
2.5	Creating and Registering Source File in Project	21
2.6	Definition of Subproject	22
2.7	Creation of Project Configuration	23
2.8	Setting Tools	25
2.9	Setting Linker Options	26
2.10	Make/Build	28
2.10.1	Making or Building of Project	29
2.11	Debugging	30
2.12	Executing Debugging Only	31
2.13	Reading SOFTUNE Project Files of Old Versions	33
2.14	Moving Project Files	35
2.15	Useful Functions	36
CHAPTER 3	Windows	37
3.1	Window Configuration	38
3.2	Tool Bar	39
3.3	Status Bar	41
3.4	Project Window	42
3.4.1	SRC Tab	43
3.4.2	ABS Tab	54
3.5	Output Window	59
3.6	Edit Window (Standard Editor)	61
3.6.1	Setting Standard Editor	64
3.7	Source Window	68
3.8	Symbol Window	73
3.9	Disassemble Window	75
3.10	Register Window	78

3.10.1	Setting Register Display	79
3.11	Memory Window	81
3.12	Local Variable Window	83
3.13	Watch Window	84
3.14	Trace Window	86
3.15	Command Window	87
3.15.1	Setting Character String Replacement	89
3.15.2	Logging	92
3.16	Object Window	93
3.17	Coverage Window	95
3.18	Performance Window	97
3.19	Sequence Window	99

CHAPTER 4 MENUS 101

4.1	Menu Configuration (Tree)	102
4.2	File Menu	105
4.2.1	New	106
4.2.2	Open	109
4.2.3	Close	111
4.2.4	Open Workspace	112
4.2.5	Close Workspace	113
4.2.6	Save	114
4.2.7	Save As	115
4.2.8	Save All	117
4.2.9	Print	118
4.2.10	Recent Text File/Recent Workspace File	119
4.2.11	Exit	120
4.3	Edit Menu	121
4.3.1	Undo, Redo	122
4.3.2	Cut, Copy, Paste, Delete	123
4.3.3	All Select	125
4.3.4	Find/Replace	126
4.3.5	Find in Files	128
4.3.6	Jump	129
4.3.7	Bookmark	130
4.3.7.1	Bookmark - Source window	131
4.3.7.2	Bookmark - Memory Window	134
4.3.7.3	Bookmark - Edit window	137
4.3.8	Previous error, Next error, Top of Error, Bottom of Error	139
4.3.9	Property	140
4.3.9.1	Property-Project Window	141
4.3.9.2	Property-Edit Window	145
4.3.9.3	Property-Source Window	146
4.3.9.4	Property-Symbol Window	147
4.3.9.5	Property-Local Window	148
4.3.9.6	Property-Watch Window	149
4.4	View Menu	150

4.4.1	Project/Output	151
4.4.2	Symbol	152
4.4.3	Assembly	153
4.4.4	Register	155
4.4.5	Memory	157
4.4.6	Local	166
4.4.7	Watch	168
4.4.8	Trace	171
4.4.9	Command	177
4.4.10	Tool Bar, Status Bar, Tab	178
4.4.11	Object	179
4.4.12	Coverage	184
4.4.13	Performance	186
4.4.14	Fonts	189
4.5	Project	191
4.5.1	Active Project	192
4.5.2	Add Project	193
4.5.2.1	Add Project - Create	194
4.5.2.2	Add Project - Existing Project	196
4.5.3	Add Member	197
4.5.3.1	Add Member - File	198
4.5.3.2	Add Member - Directory	199
4.5.4	Setup Workspace	200
4.5.5	Setup Project	201
4.5.5.1	General	204
4.5.5.2	MCU	206
4.5.5.3	Setting C Compiler Options	210
4.5.5.4	Setting Assembler Options	217
4.5.5.5	Setting Linker Options	222
4.5.5.6	Section Disposition/Connection Specifying	228
4.5.5.7	Setting Librarian Options	233
4.5.5.8	Setting Converter Options	235
4.5.5.9	Setting Debug Options	238
4.5.6	Setting Customize Build	241
4.5.7	Project Dependencies	245
4.5.8	Project Configuration	246
4.5.8.1	Project Configuration - Add and Delete	247
4.5.8.2	Project Configuration - Configuration at Build	249
4.5.9	Include Dependencies	250
4.5.10	Compile, Make, Build, and Stop	251
4.6	Debug	252
4.6.1	Run	253
4.6.2	Abort	254
4.6.3	Reset MCU	255
4.6.4	Break Point	256
4.6.4.1	Break Point Details Setting	262
4.6.5	Event	265

4.6.6	Sequence	267
4.6.7	Stack	269
4.6.8	Time Measurement	270
4.6.9	Call	271
4.6.10	Clear Call	273
4.6.11	Vector	274
4.6.12	Load Target File	276
4.6.13	Start Debug/End Debug	277
4.7	Setup	278
4.7.1	Development	279
4.7.2	Debug Environment	283
4.7.2.1	I/O Port	284
4.7.2.2	Interrupt	288
4.7.2.3	Debug Environment	289
4.7.2.4	Setup Wizard	303
4.7.3	Memory Map	305
4.7.4	FLASH Area Control	309
4.7.5	Tool	310
4.7.6	Keyboard	313
4.7.7	Editor	315
4.7.8	Error	317
4.7.9	Tool Startup	319
4.8	Window	320
4.8.1	Cascade, Vertical, Horizon	321
4.8.2	Split	322
4.8.3	Arrange Icons	323
4.8.4	Refresh	324
4.8.5	Refresh All Windows	325
4.8.6	Close All Windows	326
4.9	Help	327
4.9.1	Help Topics	328
4.9.2	Support Information	329
4.9.3	Version Information	330

CHAPTER 5 Add-in Module 331

5.1	Customize Bar	332
5.1.1	What is Customize Bar?	333
5.1.2	Customize Bar Menu	335
5.1.3	Registering in the Customize Bar	336
5.1.3.1	Registering Batch File	340
5.1.3.2	Registering Workbench Menu	342
5.1.3.3	Registering External Tool	344
5.1.4	Warning and Error Messages	346
5.1.5	Note	348

APPENDIX	349
APPENDIX A List of Register Names	350
APPENDIX B Downloading Monitor Program	351
APPENDIX C Setting LAN Interface	352
APPENDIX D Setting USB Interface	353
APPENDIX E Installing Monitor Debugger	354
APPENDIX F External I/F for Simulator	374
F.1 SSDI_Entry (Start)	376
F.2 SSDI_Init (Initialize)	377
F.3 SSDI_Ready (Ready)	378
F.4 SSDI_End (End)	379
F.5 SSDI_Reset_Event (Post Reset)	380
F.6 SSDI_Read_Event (Post Read)	381
F.7 SSDI_Write_Event (Post Write)	382
F.8 SSDI_Execute_Event (Post Execute Instruction)	383
F.9 SSDI_Interrupt_Event (Post Interrupt)	384
F.10 SSDI_Timer_Event (Post Timer)	385
F.11 SSDI_Read_Memory (Read from Memory)	386
F.12 SSDI_Write_Memory (Write to Memory)	387
F.13 SSDI_Read_Register (Read from Register)	388
F.14 SSDI_Write_Register (Write to Register)	389
F.15 SSDI_Set_Interrupt (Set Interrupt Source)	390
F.16 SSDI_Set_Timer (Set Timer)	391
F.17 SSDI_Request_Abort (Request Abort)	392
F.18 SSDI_Set_Area (Set Area)	393
F.19 SSDI_Clear_Cycle (Clear cycle count)	394
F.20 SSDI_Execute_MCU (Execute target program)	395
F.21 SSDI_GetCurrentExecuteAddr (Get the Address in Execution)	396

CHAPTER 1

OUTLINE OF SOFTUNE Workbench

This chapter gives an outline of SOFTUNE Workbench. SOFTUNE Workbench integrates language tools and debuggers into one to provide the integrated development environment that totally supports processing from programming and debugging to creation of data to be written to ROM. Language tools include a C compiler, assembler, and linkage kit, etc. Debuggers are a simulator debugger, emulator debugger, and monitor debugger.

1.1 Outline

1.2 What is SOFTUNE Workbench?

1.3 Procedure for Developing Programs with SOFTUNE Workbench

1.1 Outline

This section gives an outline of the development tools integrated by SOFTUNE Workbench.

■ Language Tools

In the past, language tools (e.g., C compiler, assembler, and linkage kit) were started and used from command lines.

However, SOFTUNE Workbench can use these tools as they are. An option setting dialog for each tool opens, thereby enabling the easy use of the tools.

■ Debuggers

SOFTUNE Workbench has integrated the simulator debugger, emulator debugger, and monitor debugger into one. The optimum debugger can be selected and used as required.

■ Others

Installing an REALOS configurator (option) enables cooperative operation without complicated setting.

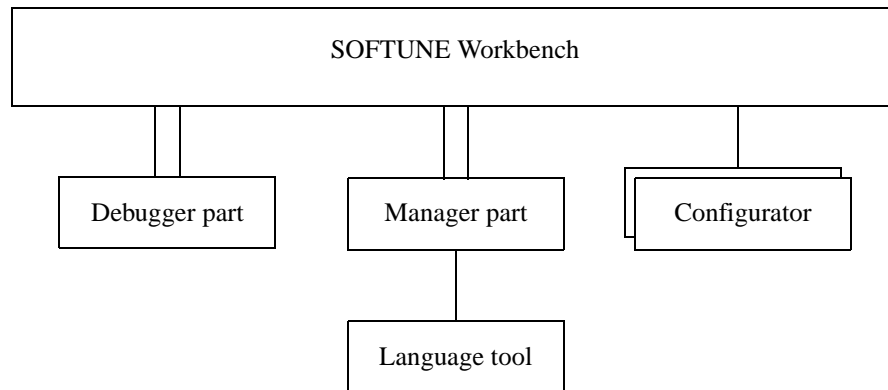
1.2 What is SOFTUNE Workbench?

This section explains the basic configuration of SOFTUNE Workbench.

■ SOFTUNE Workbench Configuration

Figure 1.2-1 shows the basic configuration of SOFTUNE Workbench.

Figure 1.2-1 Basic Configuration of SOFTUNE Workbench



As shown in Figure 1.2-1, SOFTUNE Workbench consists of three parts: body, debugger, and manager.

The debugger part contains the simulator debugger, emulator debugger and monitor debugger. These debuggers can be switched and used as required.

The manager part enables users to code and make programs without full knowledge of language tool (e.g., C compiler and assembler) start and option specification.

The configurator is not built into SOFTUNE Workbench because it is an option. Installing this option, however, enables cooperative operation on SOFTUNE Workbench.

SOFTUNE Workbench manages all processing from programming to debugging in units of projects. Projects contain all program files, options of tools (e.g., C compiler), and debugger environment setup, etc.

1.3 Procedure for Developing Programs with SOFTUNE Workbench

The procedure for developing programs with SOFTUNE Workbench consists of the followings:

1. **Setting SOFTUNE Workbench operating conditions**
 2. **Creating a project**
 3. **Creating a program source and executing make/build**
 4. **Executing the debugger**
-

■ Setting SOFTUNE Workbench Operating Conditions

When developing a program with SOFTUNE Workbench, first open the development environment setup dialog from the [Setup] - [Development] menu and set environment variables and projects. For details about how to set environment variables and projects, see Section "4.7.1 Development".

The environment variables set from this dialog are referenced by language tools such as the C compiler.

■ Creating a Project

Set information for the program to be developed in a project.

When developing a new project, open the new creation dialog from the [File] - [New] menu and select [Workspace/Project File] from the dialog. The new project creation dialog opens.

When the project has already existed, the existing project file can be opened from the [File] - [Open Workspace] menu. When using the SOFTUNE V01 or V02 project file, see Section "2.13 Reading SOFTUNE Project Files of Old Versions".

■ Creating a Program Source and Executing Make/Build

Open the new creation dialog from the [File] - [New] menu and select [Text File]. When the editor is started, write the source program and save it to the file with the [File] - [Save As] menu.

When a necessary source file is created, register it in the project with the [Project] - [Add Member] menu.

When registering the source file in the project is completed, execute "make" with the [Project] - [Make] menu or execute "build" with the [Project] - [Build] menu.

If a syntax error occurs during compilation or assembling, double-click the error display location in the output window with the left button of the mouse. The program jumps to the line where the error occurred. Correct the source file, and then reexecute the [Project] - [Make] menu.

■ Executing the Debugger

When a load module file is created, debugging can begin.

CHAPTER 2

OPERATION

This chapter describes the basic operation method and development procedure of SOFTUNE Workbench.

- 2.1 Parameters to be Entered from Dialog
- 2.2 Starting and Terminating SOFTUNE Workbench
- 2.3 Creating Workspace
- 2.4 Storing of Project
- 2.5 Creating and Registering Source File in Project
- 2.6 Definition of Subproject
- 2.7 Creation of Project Configuration
- 2.8 Setting Tools
- 2.9 Setting Linker Options
- 2.10 Make/Build
- 2.11 Debugging
- 2.12 Executing Debugging Only
- 2.13 Reading SOFTUNE Project Files of Old Versions
- 2.14 Moving Project Files
- 2.15 Useful Functions

2.1 Parameters to be Entered from Dialog

When key entry is requested from a dialog, the following four elements can be written as parameters:

- Data formula
 - Address formula
 - Identifier
 - File name specification
-

■ Data Formula

A data formula consists of a term and an operator. Data formulas comply with C language formulas. Almost C language formulas are recognized. Some points (e.g, line number and register specification) are extended. Operations involving floating-point numbers and character strings are not supported. The overflows that occur during the operation are ignored. Zero division results in an error.

■ Address Formula

An address formula is an extension of the data formula; it represents a memory location. Like the data formula, the address formula consists of a term and an operator. The terms and operators usable in address formulas are the same as those in data formulas

■ Identifier

Alphabetic characters, numbers, and "_" can be used as identifiers. Each identifier must begin with a character other than numbers. Uppercase characters are distinguished from lowercase characters or vice versa.

■ File Name Specification

File name specification complies with Windows rules.

2.1.1 Data and Address Formulas (Numerical Constant)

The SOFTUNE Workbench provides numeric constants as the terms of data and address formulas. An integer or floating-point number can be written as a numerical constant.

■ Integer

When writing an integer, specify a specifier (B', Q', D', H', 0x) representing the radix of the numerical value. If no specifier is specified, the radix specified by the default is used. The default radices are determined for each of the locations from which values must be entered. For details, see each dialog.

The numerical value representation range is from 0 to H'FFFFFFFF.

However, this range is further restricted, depending on the values to be entered.

The minus values are represented such as -D'1.

(Example)

Binary constant	B'1010
Octal constant	Q'1267
Decimal constant	D'1800
Hexadecimal constant	H'12AF or 0x12AF

Note:

No blank is allowed between a specifier and a numerical value.

■ Floating-point Number

The following two floating-point numbers are supported.

- Single-precision floating-point number (S)...float
- Double-precision floating-point number (D)...double, long double

The internal format and size comply with the floating-point number type handled by the C compiler.

$[F][-]\{.d|d[.d]\}\{S|D\}[[+|-]d]$

"d" specifies an unsigned decimal number.

Nearest value rounding applies to input values. If the represented value is not a normalized number, a warning message is displayed and the following value is input:

- When an underflow occurs The values that can be represented as unnormalized numbers are changed to unnormalized numbers. The values less than unnormalized numbers are changed to ± 0 .
- When an overflow occurs Values are changed to infinity.

CHAPTER 2 OPERATION

A floating-point number can also be specified in a hexadecimal number as follows:

H' hexadecimal - number [. {S D}]

Note:

If S and D are omitted, D is assumed.

2.1.2 Data and Address Formulas (Symbols, Line Numbers, Character Constants)

The **SOFTUNE Workbench** provides symbols, line numbers, and character constants as the terms of data and address formulas.

■ Symbol

The symbols used in the source program can be referenced as addresses; they have the type information generated by the C compiler and other accessory information. The accessory information generated by the assembler is label information. Each symbol consists of a module name, a function name, and a symbol name. Specify these names as follows:

```
[[module-name]][\function-name]\ symbol-name
```

When the source program is written in the assembly language, module-name is the name written in the operand of the [PROGRAM] statement. When it is written in the C language, module-name is the name of the source file to be compiled. Function-name is a function name written in the C language; it is valid only when the source program is written in the C language.

To distinguish a global symbol from others, write it as \symbol-name.

Table 2.1-1 shows symbol description.

Table 2.1-1 Symbol Description

Description	Contents
Symbol	Local symbol in function → static symbol in module → global symbol
\Symbol	Local symbol in the specified function in the current module
\function\symbols	Static symbol in the specified module or global symbol defined in the specified module
Module\symbols	Local symbol in the specified function in the specified module
Module\function\symbols	Symbol in class (Valid for static) Symbol in class function

■ Line Number

The line numbers to be generated by the C compiler or assembler can be used to reference addresses. For the C language, a line number indicates the starting address when one line is compiled.

```
[source file-name]$line-number
```

When referencing an address with a line number, prefix \$ to the line number. A line number can be specified only in a decimal number. Specify the line number in the following format. If the extension of the source file name is .c, line number specification can be omitted. If the source file name is not alphanumeric characters, enclose the line number in double quotes.

■ Character Constant

A character constant is the character value enclosed in a single quote; it cannot include a single quote and \. Instead of these characters (single quote and back slash), escape characters can be used as character constants. Characters that can constitute character strings can be used as escape characters.

2.1.3 Data and Address Formulas (Register Name, Flag Name)

The SOFTUNE Workbench provides register and flag names as the terms of data and address formulas.

■ Register Name, Flag Name

Register and flag names can be specified in data formulas; they represent the register values at that point of time.

Specify the register name and flag name, following %.

Usable register names differ for each MCU; see Appendix A Register Names.

2.1.4 Operators Usable in Data and Address Formulas

Table 2.1-2 lists the operators that can be used in data and address formulas and their priorities.

■ Operators Usable in Data and Address Formulas

Table 2.1-2 List of Operators Usable in Data and Address Formulas

Priority	Operator	Explanation	Type of operator
1	() [] . ->	Priority change Subscript representation Structure Structure pointer	Linear expression
2	- & ! ~ * sizeof (type)	Minus sign Address Logical NOT Bit NOT Memory indirect reference Size (byte) Type cast	Unary operator
3	* / %	Multiplication Division Remainder	Binary operator
4	+ -	Addition Subtraction	
5	<< >>	Left shift Right shift	
6	< <= > >=	Less than Less than or equal to Greater than Greater than or equal to	Binary operator (Relational operator)
7	== !=	Equal to Not equal to	
8	&	Bit AND	Binary operator
9	^	Bit EOR	
10		Bit OR	
11	&&	Logical AND	
12		Logical OR	

Note:

When the comparison result is true, the relational operator becomes H'1. When false, it becomes H'0. The SOFTUNE Workbench does not support the conditional operator (?:), comma operator (,), increment operator (++) and decrement operator (--) of C language.

2.1.5 Address Formula Specification

Address formula specification is divided into the addressing and address range specification.

■ Addressing

The addressing format is as follows:

16-bit-addressing

16-bit-addressing:Expression for addressing

■ Address Range Specification

Address range specification, representing a memory range, consists of two addressings; it has any of the following two formats:

Addressing..addressing (from starting address to ending address) Addressing..+ offset (from starting address to starting address + offset)

offset: Value relative to addressing (starting address)

■ Bit Addressing

The notation below is used to represent a bit address. Symbols for bit address attributes can be also used. Bit addressing is valid when /BIT is specified in the command qualifier.

[addressing] : bit-offset

bit-offset:Value used to specify a bit position

When addressing is omitted, address 0 is assumed.

2.1.6 File Name Specification

File name specification complies with Windows rules.

■ File Name Specification

`[drive-name:] [directory-path-name] file-name [.extension]`

When drive-name is omitted, the current drive is selected.

2.2 Starting and Terminating SOFTUNE Workbench

This section explains how to start and terminate SOFTUNE Workbench.

■ Starting SOFTUNE Workbench

With SOFTUNE V3, to start SOFTUNE Workbench, double-click the [F²MC-8L Family SOFTUNE Workbench] icon in the [SOFTUNE V3] group.

When this program is started for the first time with SOFTUNE Manager V01 or V02 installed, the dialog is displayed which asks whether or not to take over information for [Setting editor], [Setting tools], and [Setting error jump] set in the previous version. To take over the information, click the [Yes] button. Not to take over it, click the [No] button.

■ Terminating SOFTUNE Workbench

To terminate SOFTUNE Workbench, select [Exit] from the [File] menu or click the x button above and on the right of the window.

SOFTUNE Workbench cannot be terminated when compile/assemble, make, build, or tool is being executed. Be sure to terminate SOFTUNE Workbench after compile/assemble, make, build, or tool has been terminated or suspended.

2.3 Creating Workspace

Workspace is necessary to store projects in SOFTUNE Workbench to register project, follow the steps below.

■ Creating of Workspace

SOFTUNE Workbench uses the following methods to create workspace.

- Creating workspace in creating new project

When a new project is created, workspace to store the project is also created.

In this case, the name and position of workspace are the same as those of the project.

- Creating workspace in opening project

When [File]-[Open Workspace] is selected to open a project, workspace to store the project is also created.

In this case, the name and position of workspace are the same as those of the project.

If any workspace file already exists, however, it is opened instead of the project file.

- Creating blank workspace

Blank space that has no project is created. Projects must be stored separately.

In this case, projects can be stored different in name and position from workspace.

■ Creating Workspace in Creating New Project

1. Select [File]-[New].

When the [New] dialog is opened, select "Workspace/Project File" in [Type of File] and click the [OK] button.

2. Select the [Project] tab.

When the [New] dialog is opened, select the [Project] tab.

3. Select [Create New Workspace].

Click the [Create New Workspace] check button to create project. Create workspace in the same way that a new project is created.

■ Creating Workspace in Opening Project

1. Select [File]-[Open Workspace].

2. Select "Project File" in [Type of File].

When the [Open Workspace] dialog is opened, select "Project File" in [Type of File].

3. Select the project file to be opened.

Select the project file to be opened.

4. Click the [Open] button.

■ Creating Blank Workspace

1. Select [File]-[New].

When the [New] dialog is opened, select "Workspace/Project File" in [Type of File] and click the [OK] button.

2. Select the [Workspace] tab.

When the [New] dialog is opened, select the [Workspace] tab.

CHAPTER 2 OPERATION

3. Select the type of workspace.
Select [Blank Workspace] as the type of workspace.
4. Enter the workspace name.
Enter the workspace name. This name is used as a workspace file name. At default, it is also used as a workspace directory (the workspace directory can be changed).
5. Click the [OK] button.

2.4 Storing of Project

A project is necessary to develop and debug software in SOFTUNE Workbench. To store a project in workspace, follow the steps below.

■ Storing of Project

SOFTUNE Workbench uses the following methods to store a project in workspace. To store a new project as an active project in workspace.

- Storing new project in currently opened workspace

A new project is stored in currently opened workspace.

- Storing existing projects in currently opened workspace

Existing projects are stored in currently opened workspace.

■ Storing New Project in Currently Opened Workspace

1. Open workspace to store a project.
Please open workspace to be stored a project.
2. Select [Project]-[Add Project]-[New].
3. Select [Add to Current Workspace].
Click the [Add to Current Workspace] check button.
4. Select [Project Type].
Select the type of file last created in the project [Project Type]. Table 2.4-1 indicates the selectable project types and their explanation.
5. Select the chip type and target MCU.
Selectable values are indicated in the drop-down list. Select the chip and target MCU from the list.
6. Enter the project name.
Enter the project name. This name can be used as a project file name. At default it is also used as a target file name and project directory (the target file name and project directory can be changed).
7. Set project dependence.
When defining a project as a subproject in another project, place a checkmark in the [Dependence] checkbox and select the project name from the list in the [Dependence] checkbox.
8. Click the [OK] button.

"Debug", "Debug\ABS", "Debug\OBJ", "Debug\LST", or "Debug\OPT" directory is created as a sub-directory in the project directory.

Debug: A directory to store information for each project configuration. The default configuration name as a new project is created is "Debug".

ABS: Directory in which the target file is stored

OBJ: Directory in which the object file is stored

LST: Directory in which the list file is stored

OPT: Directory in which the option file is stored

When REALOS is selected as the project type, the Setup Wizard of the configurator opens. For details,

refer to the manual accompanying SOFTUNE REALOS.

Table 2.4-1 List of Project Types

Project type	Explanation
Absolute format (ABS)	An ordinary program file is created.
Relative format (REL)	A relative format file is created.
Library file	A library file is created.
REALOS (ABS)	A program that uses a real-time operating system is created.

■ Storing Existing Projects in Workspace

1. Open workspace to store a project.
Open workspace to store a project.
2. Select [Project]-[Add Project]-[Project].
3. Select the project to be stored.
Select the project to be stored in the [Add Project] dialog.
4. Set project dependence.
When defining a project as a subproject in another project, place a checkmark in the [Dependence] checkbox and select the project name from the list in the [Dependence] checkbox.
5. Click the [Open] button.

■ Setting Active Project

The active project is a project that undergoes [Make], [Build], [Compile/Assemble], [Start Debug] and [Include Dependence] in the menu. [Make], [Build], [Compile/Assemble], and [Include Dependence] affects the subprojects in the active project.

To set an active project, select [Project]-[Set Active Project]. When the submenu is displayed, select the name of a project to be made active from the submenu.

■ Deleting Project Stored in Workspace

Select the project to be deleted in the SRC tab of project window. Select [Delete Project] in the short-cut menu.

The specified project is deleted from workspace, but the project file itself is not deleted.

If the deleted project is used as the subproject in the project within workspace, the project dependence is also deleted.

Note:

If SOFTUNE REALOS is not installed, REALOS is not displayed as the project type.

2.5 Creating and Registering Source File in Project

This section explains the procedure for creating a new source file with SOFTUNE Workbench and registering the file in the project.

■ Creating the New Source File

1. Select the [File]-[New] menu.
When the [New] dialog opens, select [Text File] from [Types of Files], then click the [OK] button.
2. Select the [File]-[Save As] menu.
When the [Save As] dialog opens, select [Text File] from [Types of Files], then click the [OK] button.
When the file dialog for specifying the directory to which the created file is to be saved and the file name opens, select the directory, specify the file name, then click the [Save] button.

■ Registering the Created File in the Project

Select the [Project] - [Add Member] - [File] menu. The File dialog is opened to select the file to be added to the member. Select the created source file, followed by the folder inserted into the SRC tab of project window, and click the [Open] button.

The file is stored in the project and its name is displayed in the source file category in the SRC tab of project window.

■ Storing Created File with Directory in Project

Select the [Project]-[Add Member]-[Directory] menu. The "Add Member-Directory" dialog is opened to select the folder to be added to the member. Select the directory having the created source file, followed by the folder to be inserted into the SRC tab of project window, and click the [OK] button. The file and directory are stored in the project and the file and folder below the specified directory are displayed in the specified folder in the SRC tab of project window.

The type of file to be stored can be restricted by setting [Type of File] in the dialog.

■ Deleting Files Stored in the Project

Select a file(s) to be deleted in the SRC tab of project window. (Multiple file can be selected.) Select "Delete" from the short-cut menu. The selected file is deleted from the project member, but the file itself is not deleted. Users cannot delete files in the [Dependencies] category and [Debug] category files.

2.6 Definition of Subproject

This section explains how to define a subproject.

■ Definition of Subproject

The subproject is a project on which other projects depend.

SOFTUNE Workbench uses the following methods to define a subproject.

- Defining project as subproject in storing it

When created, a new project is defined as a subproject in another project. For the setting method, see Section "2.4 Storing of Project".

- Defining subproject between existing projects

A subproject is defined between projects in workspace.

Another project is defined as a subproject in the subproject in the parent project. Such a recurrent definition that the parent project itself serves as a subproject is impossible.

■ Defining Subproject between Existing Projects

1. Select [Project]-[Dependence].
2. Select the parent project in which a subproject is defined.
When the [Dependence] dialog is opened, select the name of the parent project in which a subproject is defined from the [Project Name] box.
3. Select the project that is defined as a subproject.
Check the project that is defined as a subproject from those in the [Dependent Project] list.
4. Click the [OK] button.

■ Deleting Subproject from Project

1. Select [Project]-[Dependence].
2. Select the parent project from which a subproject is deleted.
When the [Dependence] dialog is opened, select the name of the parent project from which a subproject is deleted from the [Project Name] box.
3. Select the subproject to be deleted.
Deselect the subproject that is deleted from the [Dependent Project] list.
4. Click the [OK] button.

2.7 Creation of Project Configuration

This section explains how to create a project configuration.

■ Creation of Project Configuration

The project configuration is a series of settings for specifying the characteristics of the target file. By creating a new project configuration, two or more tool settings can be stored in the project.

When a new project is created, the project configuration is created under a default name of "Debug".

In SOFTUNE Workbench, the project configuration is created as follows.

- Creating project configuration on settings of existing project configuration

A new project configuration is created on the settings of the selected existing project configuration. In the new project configuration, the same files as those in the original project configuration are always used.

■ Creating Project Configuration on Settings of Existing Project Configuration

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project to which a project configuration is added.
When the [Add and Delete Project] dialog is opened, select the project to which a project configuration is added.
3. Click the [Add] button.
Click the [Add] button. The [Add Project Configuration] dialog is opened.
4. Enter the project configuration name.
Enter the unique name of a new project configuration. The characters that can be used to form a name are "a to z", "A to Z", "0 to 9" and "_".
5. Select the project configuration to which settings are copied.
Select the initial settings of a project configuration to be added. The selected settings of the project configuration (such as tool options, file configurations, and configurations of subprojects to be build) are copied as they are.
6. Click the [OK] button.
Click the [OK] button in the [Add Project Configuration] dialog and the [OK] button in the [Add and Delete Project Configuration] dialog.

■ Setting Active Project Configuration

The active project configuration is a project configuration that undergoes [Make], [Build], [Compile/Assemble], [Start Debug], and [Include Dependence] at default.

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project configuration that is made active.
When the [Add and Delete Project] dialog is opened, select the name of the project configuration that is made active.
3. Click the [Active] button.
Click the [Active] button. The specified configuration and its project become active.
4. Click the [OK] button.

■ Deleting Specific Project Configuration from Project

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project configuration that is deleted from the project.
When the [Add and Delete Project] dialog is opened, select the project configuration name to be deleted.
3. Click the [Delete] button.
Click the [Delete] button. The specified project configuration is deleted. When all project configurations in a project are deleted, the project itself is also deleted.
4. Click the [OK] button.

2.8 Setting Tools

When make or build is executed by SOFTUNE Workbench, appropriate options must be set in such tools as a compiler, assembler and linker. Set these options as follows:

■ Select the [Project]-[Setup Project] Menu

The [Set Project] dialog is opened. The option selected in the [Set Project] dialog box is applicable to two or more projects. The applicable project configuration can be limited. For example, the settings of project configurations A and B can be changed. The same setting can be also specified for all project configurations.

Settings must be made for each project. To make settings for a project, select a project configuration in the [Setting Target] dialog box and specify the project in the tree view that appears. After selecting the project, select the tab for the tool you want to set up.

When the compiler, assembler and linker/librarian are selected, the category can be selected in the top tab of the setting dialog for each tool. After the drop-down list is opened, select a category. When a category was selected, the contents in the display are changed and the options included in each category can be set.

In most cases, compiler and assembler options need not be set except when output of list file and make or build is executed under specific conditions. Set only linker options. For how to set linker options, see Section "2.9 Setting Linker Options".

■ Click the [OK] Button to Complete Tool Setting

When setting all necessary tool options is completed, click the [OK] button. All the set options are registered in the project; they become valid when make or build is executed.

Clicking the [Cancel] button cancels all the set options.

Note:

When the [Update] button is clicked during tool option setting, the previously set options cannot be restored.

2.9 Setting Linker Options

When creating a program with SOFTUNE Workbench, be sure to set a memory map with a linker option.

■ Automatic Setting of Linker Options

In SOFTUNE Workbench, the following linker options are automatically set on the basis of information on the MCU selected when a new project is created;

- Specify the internal ROM/RAM address of the MCU in the memory area option.

Internal ROM is output under an area name of "_INROMxx" and internal RAM under an area name of "_INRAMxx" (where x is numbered consecutively starting with 01).

- Set the auto disposition mode to mode 2 (optimum auto disposition by linker).

When creating a program in a mode other than the single-chip mode or when customizing the disposition of sections, set the [Disposition/Connection] option as the linker option.

■ Setting of Linker Options

In SOFTUNE Workbench, specifying memory mapping is basic to the disposition of each section in the memory area. Therefore, set a memory area and set each section in the memory area.

■ Setting of Memory Area

Enter a ROM/RAM area name, start address and end address, select an area attribute (ROM or RAM), then click the [Setup] button. This setting is displayed in the ROM/RAM area list. In the ROM/RAM area, assign a unique ROM/RAM name so that it does not match other area names.

An easy program consists of a ROM area and a RAM area, but in a complicated program, several areas may be specified. The number of areas that can be set is not limited; set all the areas necessary to configure the memory map of the program to be developed.

In Auto Disposition (Mode 2), the linker automatically allocates sections unspecified for allocation in a ROM/RAM area.

The linker searches an available ROM/RAM area beginning at the top of the [ROM/RAM Area List]. Click the [Up] button or the [Down] button to change the desired number.

■ Setting of Sections

Selecting the area from the ROM/RAM area list and clicking the [Setup Section] button open the [Setup Section] dialog, enabling the sections to be allocated to the selected area. When selecting an area, click the start address of the area.

When the [Setup Section] dialog opens, specify section names in the order the sections are allocated to the area. Specify section names one by one. When section name specification is completed, click the [Setup] button to register the section names in the section name list.

When make or build is executed, the sections are allocated to the area in the order the section names were registered in the section name list.

When setting the sections to be allocated to one area is completed, click the [OK] button to return the linker option setting dialog. Also set other areas in the same way.

Reference:

Allocating sections to several areas can be continuously set by changing ROM/RAM area name display in the uppermost part of the [Setup Section] dialog. The linker option setting dialog need not be returned each time sections are allocated to an area.

2.10 Make/Build

SOFTUNE Workbench can create a program in two methods: make and build.

■ Make

Compiles or assembles only the modified source file and then links all objects to the library to generate an object program. SOFTUNE Workbench recognizes the dependency of the include files registered in the [Dependencies] category of the SRC tab of project Window to compile or assemble the source file.

Use the [Project]-[Make] menu to execute make.

■ Build

Not only modified source file, but compiles or assembles all the source files registered in the project and then links all objects to the library to generate a target file.

Use the [Project]-[Build] menu to execute build.

■ Stop

Stop is the function that forcibly suspends processing during make, build, compilation, or assembling.

Execute stop with the [Project]-[Stop] menu during make, build, compilation, or assembling.

2.10.1 Making or Building of Project

SOFTUNE Workbench enables making or building for each project configuration.

■ Making or Building of Project

[Make] or [Build] in the menu applies to the active configuration of an active project. If a subproject is defined, priority is given to making or building of the subproject.

For details about how to change the active project and active configuration, see Section "2.4 Storing of Project" and Section "2.7 Creation of Project Configuration".

■ Making or Building Specified Project

Select the project to be made or built in the SRC tab of project window. Select [Make] or [Build] in the short-cut menu. The active configuration of the specified project is made or built. If a subproject is defined, priority is given to making or building of the subproject.

■ Changing Subproject Configuration at Making or Building

1. Select [Project]-[Project Configuration]-[Configuration at Build].
2. Select the parent project and configuration.
When the [Set Configuration at Build] dialog is opened, select the project to be set from the [Project] box and its configuration. The configuration of a subproject to be made or built is displayed.
3. Select the configuration of a subproject.
Select the configuration to be made or built from [Configuration of Subproject at Make/Build].
4. Click the [OK] button.

2.11 Debugging

The absolute file created as a result of normal termination of make/build can be debugged immediately after SOFTUNE Workbench has migrated to the debug session.

■ Migrating SOFTUNE Workbench to Debug Session

To enable SOFTUNE Workbench to debug the absolute file, migrate it to the debug session.

To migrate SOFTUNE Workbench to the debug session, select [Start Debug] from the [Debug] menu.

-First debugging after project creation

Setup Wizard for debuggers is started. Set the type of the debugger and others (See Section "4.7.2.4 Setup Wizard").

-Second or subsequent debugging after project creation

Start SOFTUNE Workbench in the debugger mode that has been already set. To change the type of the debugger, select the [Debug]-[End debug] menu to terminate debugging once, and then change [Project] - [Project Setup] menu.

■ Loading the Target Program

When SOFTUNE Workbench enters the debug session, select [Load target program] from the [Debug] menu to load the target program. The created program is loaded to the debugger and all debugging preparations are completed now.

■ Operating the Debugger

For how to operate the debugger, see Chapter 3 Windows and Chapter 4 Menus in this manual.

For debugger commands and debugger output error messages, refer to the SOFTUNE Workbench Command Reference Manual.

For the debugger functions (MCU common function, MCU chip dependency function), refer to the SOFTUNE Workbench Users Manual.

2.12 Executing Debugging Only

SOFTUNE Workbench can be used as the conventional debuggers.

■ Creating a Project

In SOFTUNE Workbench, projects are a basis of all work. This is not an exception also at debugging. For this reason, executing debugging only requires the creation of a project for debugging.

First, create the project for debugging and the workspace which stores the project in the following procedure:

1. Select [New] from the [File] menu.
2. Select [Project/Workspace File] from the file open dialog.
 - Select the absolute format (ABS) from the new creation dialog.
 - Specify a project name.
 - Specify a project directory.
 - Select a target MCU name and chip type.

■ Setting of Workspace

Perform setting common to projects to be stored in workspace.

1. Select [Set Workspace] from the [Project] menu.
2. When the [Set Workspace] dialog is opened, perform the following settings:
 - Debug when workspace opened: Start debugging.
 - Save setup information: Save.

■ Settings Related to the Debugger

1. Select [Project Setup] from the [Project] menu.
2. When the setup dialog opens, open the [Debug] tag and select [Setup] category:
3. Set a setup name.

A project name is set both in [Setup Name List] and [Valid Setup Name] as the default setup name. Usually, setup names are identified by the type of the debugger to be used. However, if only one debugger is used, the default name may be used as it is.

When the default name is used, select the default name already set in [Setup Name List], then click the [Change Setup] button. When another name is used, specify [Setup Name], then click the [Add Setup] button.

■ Setup Wizard

Clicking the [Add Setup] or [Change Setup] button starts the Setup Wizard for debuggers. Once Setup Wizard has been started, set items according to instructions from Setup Wizard. (For how to set items with Setup Wizard, see Section "4.7.2.4 Setup Wizard".)

When all settings with Setup Wizard are completed, click the [Complete] button.

When the [Project Setup] dialog is redisplayed, click the [OK] button.

When all the above steps are completed, save the project, then close it once.

■ Starting Debugging

When steps from [Creating a project] to [Setup Wizard] are completed, open the project. SOFTUNE Workbench automatically migrates to the debug session, enabling the immediate start of debugging.

Select [Open] from the [File] menu and specify the load module file to load the target program.

2.13 Reading SOFTUNE Project Files of Old Versions

The SOFTUNE project files of old versions can be read.

■ Procedure

- For project files of SOFTUNE Workbench V3

1. Select [Open Workspace] from the [File] menu.
2. Select 'Project file' from File Type and specify the project file made by the early version of SOFTUNE Workbench. If the specified file is one made by the early version of SOFTUNE Workbench, a dialog asking whether to convert the file to a workspace project format is opened.

[Yes] button: The project file is converted and opened in the workspace project format.

[No] button: The project file is not converted and is opened in the old project format. In this case, some functions cannot be used. For details of functions that can be used in the old project format, refer to Section 1.2 Management Function for Project of SOFTUNE Workbench Users Manual.

[Cancel] button: Opening of the project file is canceled.

- For project files of SOFTUNE Manager V01/V02

1. Select [Open Workspace] from the [File] menu.
2. Select 'Project file' from File Type and specify the project file made by SOFTUNE Manager. If the specified file is one made by SOFTUNE Manager, a dialog asking whether to convert the file to a workspace project format is opened.
3. Click the [OK] button to start conversion.
Click the [Cancel] button to cancel opening of the project file.
4. When the New Project Creation Window opens, set the chip type and target MCU, then click the [OK] button.
5. When conversion is completed, the dialog showing the end of conversion opens. Click the [OK] button to close the dialog.

■ Backup File

In SOFTUNE Workbench, when a project file is converted to a workspace project format, a backup file is made automatically. The extension of backup file varies with the type of project file. The method for opening the backup project file is also different depending on the extension.

● SOFTUNE Workbench V3

Old project file (.prj) → .p03

Old option data file (.dat) → .d03

● SOFTUNE Manager V01/V02

Old project file (.prj) → .V01

● SOFTUNE Workbench V3 (.p03)

- Change the extension of the backup project file (.p03) and option data file (.d03) to '.prj' and '.dat,' respectively.

● SOFTUNE Manager V01/V02 (.v01)

- Change the extension of the backup project file (.v01) to '.prj.'

Note:

- Tool options are not passed to projects of SOFTUNE Manager V01/V02. Reset these options after read has terminated.
 - Be sure to reset "User Include File Directory" set by "Set Environment Variable" of SOFTUNE Manager V01/V02 as the "Include Path" option of the C compiler or assembler. Also be sure to reset "Library Search Path" set by "Environment Variable Setup" of SOFTUNE Manager V01/V02 as the "Library Search Path" option of the linker.
 - If the workspace file having the same name as that of the specified project file is in the same directory, the workspace file is opened instead of the project file, and no project file is converted.
-

2.14 Moving Project Files

This section explains how to move a project file to another directory or a personal computer.

■ Procedure

1. Set the path to the member stored in the project file to the relative path from the project file.
In SOFTUNE Workbench, files in the same drive as that of the project file are usually stored in the relative path. To check whether the files are stored in the relative path, check File Property in the SRC tab of project window. For the file property, see Section "4.3.9 Property".
2. Set the path to the target file directory, object file directory and list file directory to the relative path from the project file.
In SOFTUNE Workbench, when a new project is created, the output directory is set to the relative path from the project file. To make a change and check, open the [Set Project] dialog. For details, see Section "4.5.5 Setup Project".
3. Set the directories such as the include path and library path specified in the tool option to the relative path or macro description.
For the macro description, see Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE Workbench User's Manual. For the tool option, see Section "4.5.5 Setup Project".
4. After the completion of the setting in steps 1 to 3, save the project.
5. Move the project without changing the directory structure in steps 1 to 3.

2.15 Useful Functions

This section explains some useful functions to use in the SOFTUNE Workbench.

■ Tab Display of the Windows

In the SOFTUNE Workbench, the displayed windows are tabbed.

This makes it easier to find the desired window quickly when multiple windows are displayed.

■ Drag & Drop between Windows

SOFTUNE Workbench supports the drag & drop function between windows in the debugger. The followings are the windows that currently support the drag & drop function.

Please refer to the description of each window for the operation by the drag & drop.

- Project window
- Source window
- Memory window
- Disassemble window
- Watch window
- Coverage window

■ Confirmation of the Setting

SOFTUNE Workbench has various dialogs to set up functions. Each dialog requires you to click [Set] button to take effect after you set up. If you attempt to close the dialog without clicking the [Set] button, a dialog for confirmation will be displayed so that you will not fail to set up.

CHAPTER 3

Windows

This chapter explains SOFTUNE Workbench windows.

- 3.1 Window Configuration
- 3.2 Tool Bar
- 3.3 Status Bar
- 3.4 Project Window
- 3.5 Output Window
- 3.6 Edit Window (Standard Editor)
- 3.7 Source Window
- 3.8 Symbol Window
- 3.9 Disassemble Window
- 3.10 Register Window
- 3.11 Memory Window
- 3.12 Local Variable Window
- 3.13 Watch Window
- 3.14 Trace Window
- 3.15 Command Window
- 3.16 Object Window
- 3.17 Coverage Window
- 3.18 Performance Window
- 3.19 Sequence Window

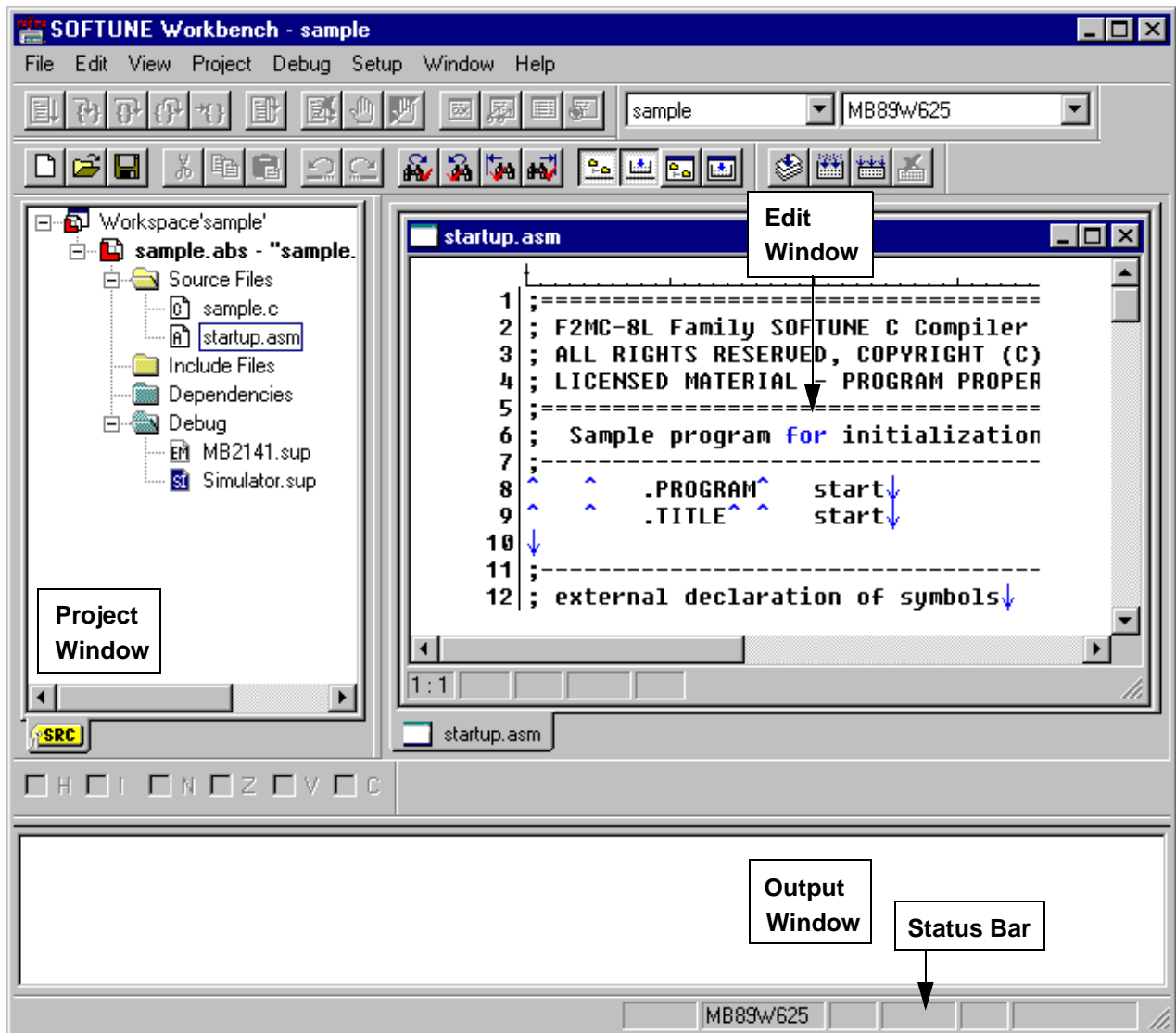
3.1 Window Configuration

Each SOFTUNE Workbench window consists of a menu bar, tool bar, window screen, and status bar. Menus are explained in Chapter 4.

■ Main Window

Figure 3.1-1 shows the SOFTUNE Workbench Main Window. As shown in this figure, child windows (e.g., project window and output window) and the tool bar can be docked with the Main Window and displayed.

















Figure 3.1-1 SOFTUNE Workbench Main Menu












3.2 Tool Bar

Command buttons to be used often are assigned to the tool bar for each group. The groups that can be selected and the command buttons in the groups are shown below.

■ Common Bar

	New
	Open
	Save
	Cut
	Copy
	Paste
	Undo
	Redo
	Jump to Next error
	Jump to Previous error
	Jump to Top error
	Jump to Bottom error
	Project window Docking
	Output window Docking
	Open Project window
	Open Output window

■ Find





	Find Next
	Find Previous
<input type="text"/>	Find
	Find in files
	Setup/cancel bookmark
	Next bookmark
	Previous bookmark
	Next bookmark in the current window
	Previous bookmark in the current window
	Clear all bookmarks in the current window

■ Project Bar













The active project and its active configuration name are set and displayed.






■ Build Bar

- | | |
|---|------------------|
|  | Compile/Assembly |
|  | Make |
|  | Build |
|  | Stop Make/Build |

■ Debug Bar

- | | |
|---|-----------------------|
|  | Go |
|  | Step In |
|  | Step Over |
|  | Step Out |
|  | Run Until Cursor |
|  | Reset MCU |
|  | Abort |
|  | Break Point Set/Reset |
|  | Register window |
|  | Watch window |
|  | Memory window |
|  | Disassembly |

■ Flash (MB2146-09)

- | | |
|---|-----------------------|
|  | Download Flash Memory |
|  | Upload Flash Memory |
|  | Erase Flash Memory |

■ Flag Bar

Display and setting of MCU condition flag status



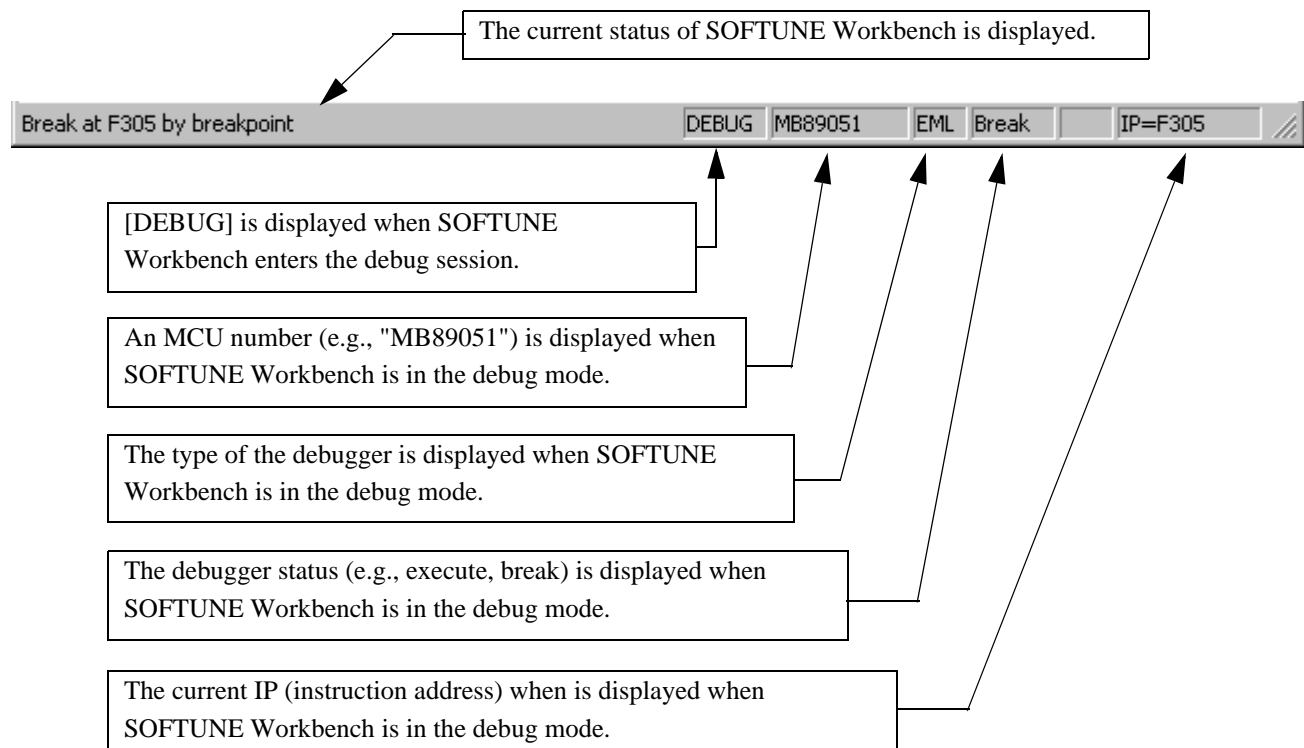
3.3 Status Bar

The current status of SOFTUNE Workbench is displayed. The status bar is displayed in the last part of the main window.

■ Status Bar

The status bar has the areas in which information is displayed. The information and the areas in which is displayed are simply shown in Figure 3.3-1.

Figure 3.3-1 Status Bar



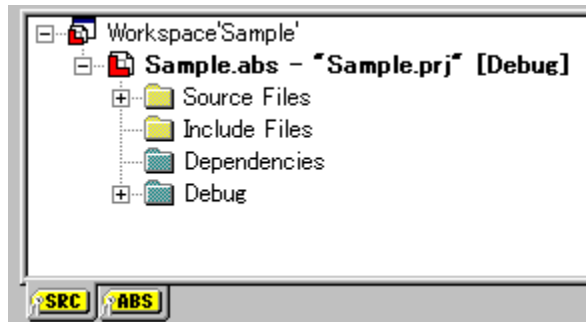
3.4 Project Window

The Project Window displays information about the project and load module.

■ Project Window

Figure 3.4-1 shows an example of the project window.

Figure 3.4-1 Project Window



To select the window display contents, use the tab on the bottom of the project window. The project window has the following tabs.

- SRC tab

Displays information on the project. Refer to Section "3.4.1 SRC Tab" for details.

- ABS tab

Displays information on the source file acquired from the debug information. Refer to Section "3.4.2 ABS Tab" for details.

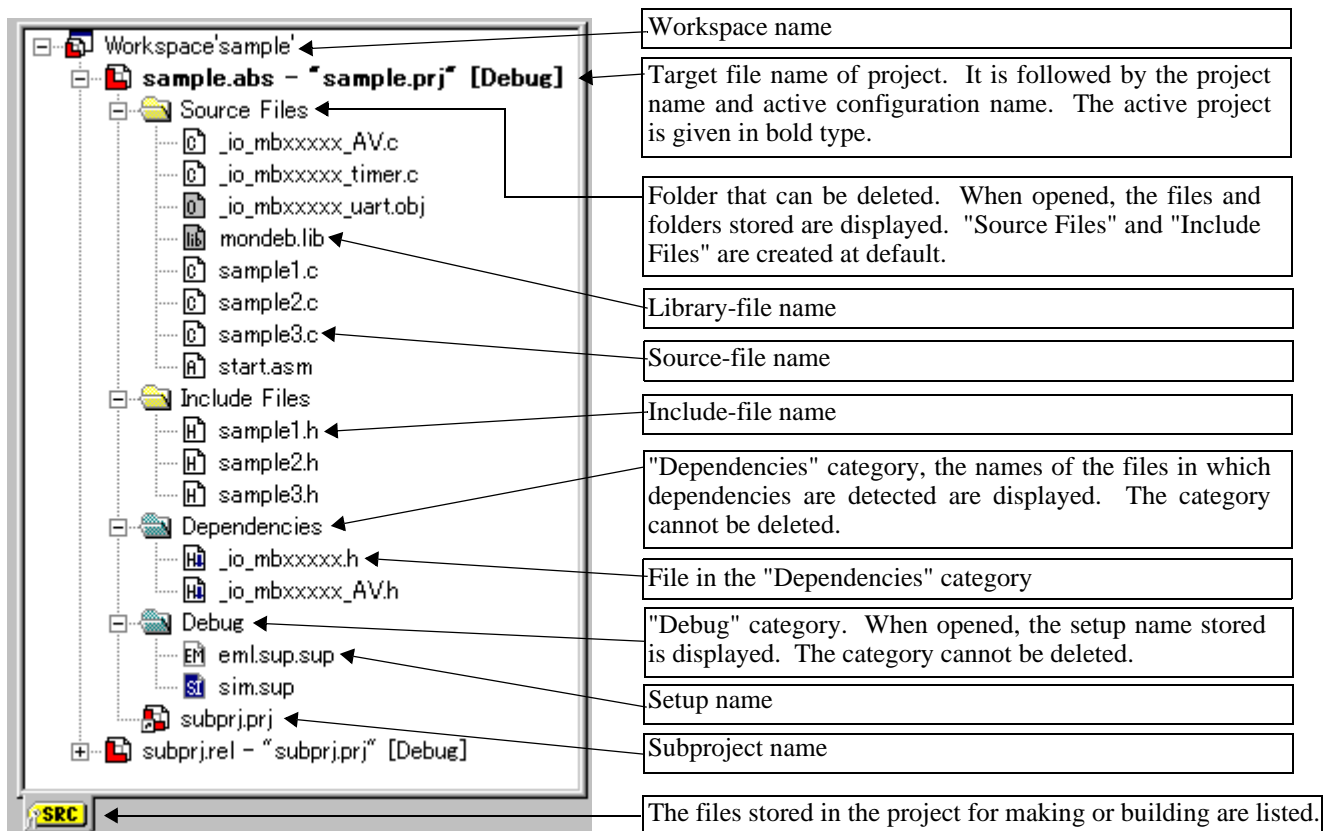
3.4.1 SRC Tab

The name of the current project and the file names registered in the project are displayed in the tree view format.

■ SRC Tab

Figure 3.4-2 shows examples of displayed contents of the SRC tab.

Figure 3.4-2 SRC Tab



■ Function

- Display the workspace name
- Display all projects stored in workspace
- Display the subproject

The subproject in the project is displayed below the parent project.

- Display of the target file name
- Display of all the source file names registered in the project

When a source file name is double-clicked, the editor opens to enable the editing of the source file.

- Display of the include files that are in dependency

When an include file name is double-clicked, the editor opens to enable the editing of the include file.







- Displaying Debugger setup name

When the Debugger setup name is double-clicked, debugging is started based on the setup information.

- Drag and drop

The member in the SRC tab can be moved and stored from the explorer.

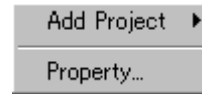
■ SRC Tab Bitmap Image List

	Workspace file name
	Target file name
	Subproject name
	Folder
	Category
	C source file
	C source file (not subject to making or building)
	C source file in which individual options specified
	C source file in which individual options specified (not subject to making or building)
	Assembler source file
	Assembler source file (not subject to making or building)
	Assembler source file in which individual options specified
	Assembler source file in which individual options specified (not subject to making or building)
	Include file
	Include file (with dependence)
	Library file
	Library file (not subject to making or building)
	Object file
	Object file (not subject to making or building)
	Relative format file
	Relative format file (not subject to making or building)
	REALOS configuration file
	Other user's registration files
	Simulator Debugger setup name
	Valid setup name (for Simulator Debugger)
	Emulator Debugger setup name
	Valid setup name (for Emulator Debugger)
	Monitor Debugger setup name
	Valid setup name (for Monitor Debugger)

■ Short-Cut Menus (Click the Right Button of the Mouse on a Workspace Name)

Figure 3.4-3 shows a short-cut menu.

Figure 3.4-3 Short-Cut Menu on a Workspace Name



- Add Project

When selected, the following two menus are displayed.

- New

A new project is added to workspace (See Section "4.5.2 Add Project").

- Existing Project

An existing project is added to workspace (See Section "4.5.2 Add Project").

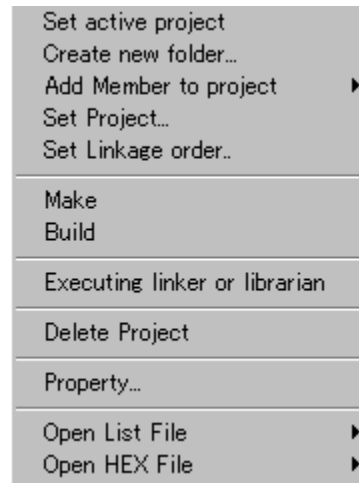
- Property

Information on the workspace file is displayed (See Section "4.3.9 Property").

■ Short-Cut Menus (Click the Right Button of the Mouse on a Target Name)

Figure 3.4-4 shows a short-cut menu.

Figure 3.4-4 Short-Cut Menu on a Target Name



- Set Active Project

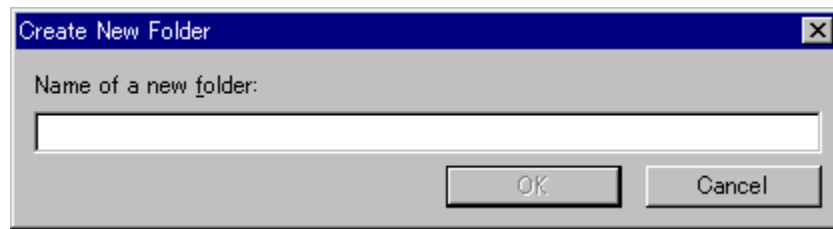
The currently selected project is set in the active project in workspace.

- Create New Folder

When selected, the [Create New Folder] dialog (Figure 3.4-5) is opened. Enter the folder name and click the [OK] button.

When the folder name is entered, the folder is inserted into the hierarchy immediately below the project. The folders are listed in the order in which they are created.

Figure 3.4-5 Create New Folder Dialog



- Add Member to Project

When selected, the following two submenus are displayed.

- File

A member is added in files to the currently selected project (See Section "4.5.3 Add Member").

At default, a member is inserted into the hierarchy immediately below the project.

- Directory

A member is added in directories to the currently selected project (See Section "4.5.3 Add Member").

At default, a member is inserted into the hierarchy immediately below the project.

- Set Project

A project is set (See Section "4.5.5 Setup Project").

- Set Linkage Order

When selected, the [Set linkage Order] dialog (Figure 3.4-6) is opened. The files displayed in the [Link Order] box are linked from top to down. To change the link order, select the configuration to be set from the [Setting Target] box, followed by the file name, and use the [Up] or [Down] button to move the file to a desired position.

The [Export] dialog (Figure 3.4-7) allows the current order to affect other configurations.

The files not subject to link are displayed in gray.

Figure 3.4-6 Set Linkage Order Dialog

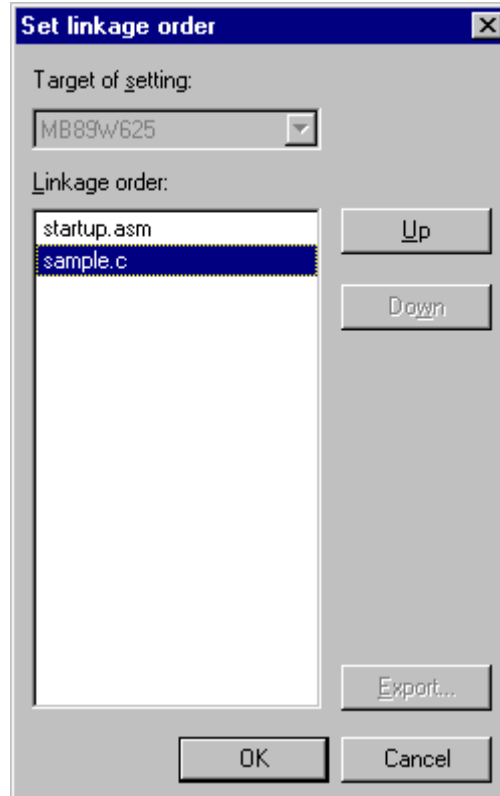
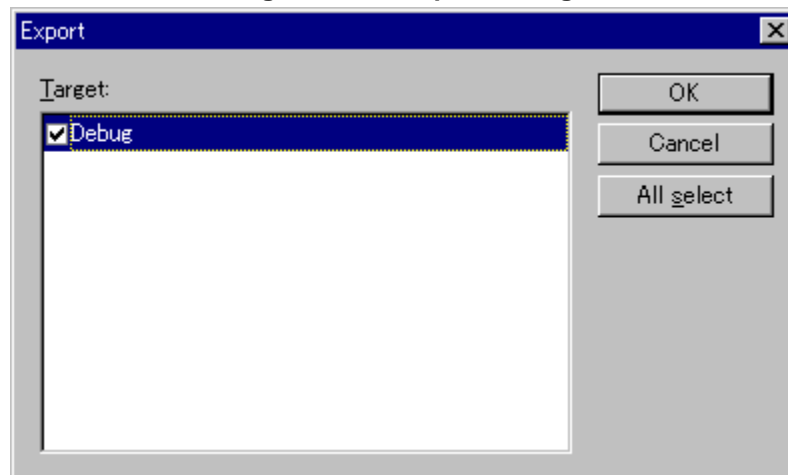


Figure 3.4-7 Export Dialog



- Make/Build

Making or building is performed in the active configuration of the currently selected project to create a target file.

- Executing linker or librarian

The linker or librarian is started in the active configuration of the currently selected project to create a target file.

- Delete Project

The specified project is deleted from workspace. The project file itself is not deleted.

- Property
File information is displayed. (See Section "4.3.9 Property")
- Open List File
Select the list file you want to open, from the submenu.
If the list file you want to open is not created, no submenu item can be selected.
- Open HEX File
Select the HEX file you want to open, from the submenu.
If the HEX file you want to open is not created, no submenu item can be selected.

■ Short-Cut Menu (Click the Right Button of the Mouse on a Subproject Name)

Figure 3.4-8 shows a short-cut menu.

Figure 3.4-8 Short-Cut Menu on a Subproject Name

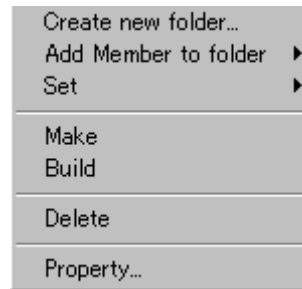


- Delete Subproject
The subproject in the specified project is deleted. To define a subproject again, see Section "4.5.7 Project Dependencies".

■ Short-Cut Menus (Click the Right Button of the Mouse on a Folder)

Figure 3.4-9 shows a short-cut menu.

Figure 3.4-9 Short-Cut Menu on a Folder



- Create New Folder
When selected, the [Create New Folder] dialog (Figure 3.4-5) is opened. Enter the folder name and click the [OK] button.
When the folder name is entered, a folder is inserted into the selected folder.
The folders are listed in the order in which they are created.
- Add Member to Folder
When selected, the following two submenus are displayed:
 - File
A member is added in files to the currently selected project (See Section "4.5.3 Add Member").
At default, a member is inserted into the selected folder.
 - Directory
A member is added in directories to the currently selected project (See Section "4.5.3 Add Member").
At default, a member is inserted into the selected folder.

- Set

When selected, the following two submenus are displayed:

- Individual Setting

A project is set (See Section "4.5.5 Setup Project").

- Return to Common Setting

All individual options in the specified folder are returned to common options.

- Make/Build

Making or building is performed in the active configuration of the currently selected project to create a target file.

- Delete

The selected folder and all files in the folder are deleted from the project. The files themselves are not deleted.

If the files (RCF files) that cannot be deleted are included, the folder is not deleted.

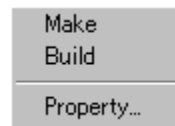
- Property

Information on folders is displayed (see Section "4.3.9 Property").

■ Short-Cut Menu (Click the Right Button of the Mouse on a "Dependencies" Category)

Figure 3.4-10 shows a short-cut menu.

Figure 3.4-10 Short-Cut Menu on a "Dependencies" Category



- Make/Build

Making or building is performed in the active configuration of the currently selected project to create a target file.

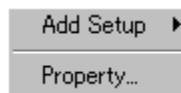
- Property

Information on folders is displayed (see Section "4.3.9 Property").

■ Short-Cut Menu (Click the Right Button of the Mouse on a "Debug" Category)

Figure 3.4-11 shows a short-cut menu.

Figure 3.4-11 Short-Cut Menu on a "Debug" Category



- Add Setup

When selected, the following two submenus are displayed:

- New

When selected, the [Create New Setup] dialog (Figure 3.4-12) is opened. Enter the setup name and click the [OK] button.

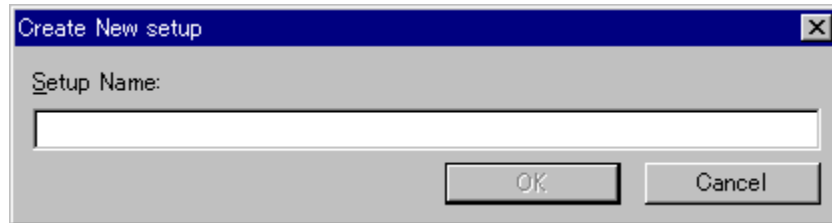
When the setup name is entered, the setup wizard is started. For details about the setup wizard, See Section "4.7.2.4 Setup Wizard".

- Reference

When selected, the [Create New Setup] dialog (Figure 3.4-12) is opened. Enter the setup name and click the [OK] button.

When the setup name is entered, the [Select File] dialog is opened. Select the file and click [Open] button. Information is read from the specified file.

Figure 3.4-12 Create New Setup Dialog



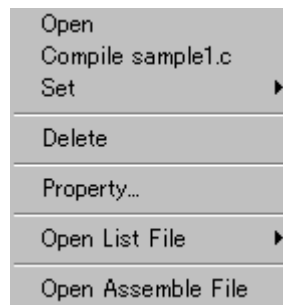
- Property

Information on folders is displayed (see Section "4.3.9 Property").

■ Short-Cut Menus (Click the Right Button of the Mouse on a Source File Name)

Figure 3.4-13 shows a short-cut menu.

Figure 3.4-13 Short-Cut Menu on a Source File Name



- Open

When SOFTUNE Workbench is in the debug session, the specified source file is displayed in the Source Window.

When SOFTUNE Workbench is not in the debug session, the specified source file is displayed in the edit window.

- Compile (Assemble)

The specified source file is automatically judged whether it is the C or assembler source and then compiled or assembled.

- Set

When selected, the following two submenus are displayed:

- Individual Setting

A project is set (See Section "4.5.5 Setup Project").

- Return to Common Setting

All individual options in the specified folder are returned to common options.

- Delete

The specified file is released from the project member. The file itself is not deleted.

- Property

File information is displayed. (See Section "4.3.9 Property").

- Open List File

Select the list file you want to open, from the submenu.

If the list file you want to open is not created, no submenu item can be selected.

- Open Assemble file

This menu is opened when a C source file is selected.

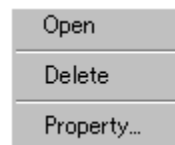
Of the specified C source file, the assembler source file output by a compiler is opened.

No file can be selected when not compiled.

■ Short-Cut Menus (Click the Right Button of the Mouse on an Include File and Other User's Registration File)

Figure 3.4-14 shows a short-cut menu.

Figure 3.4-14 Short-Cut Menu on a Include File Name



- Open

When SOFTUNE Workbench is in the debug session, the specified include file is displayed in the Source Window.

When SOFTUNE Workbench is not in the debug session, the specified include file is displayed in the edit window.

- Delete

The specified file is released from the project member. The file itself is not deleted.

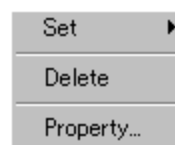
- Property

File information is displayed. (See Section "4.3.9 Property").

■ Short-Cut Menus (Click the Right Button of the Mouse on Library File, Object File and Relative Format File)

Figure 3.4-15 shows a short-cut menu.

Figure 3.4-15 Short-Cut Menu on a Library File Name



- Set

When selected, the following two submenus are displayed:

- Individual Setting

A project is set (See Section "4.5.5 Setup Project").

- Return to Common Setting

All individual options in the specified folder are returned to common options.

- Delete

The specified file is released from the project member. The file itself is not deleted.

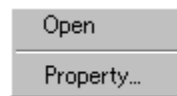
- Property

File information is displayed. (See Section "4.3.9 Property").

■ Short-Cut Menus (Click the Right Button of the Mouse on a File in the "Dependencies" Category)

Figure 3.4-16 shows a short-cut menu.

Figure 3.4-16 Short-Cut Menu on a File in the "Dependencies" Category



- Open

When SOFTUNE Workbench is in the debug session, the specified source file is displayed in the Source Window.

When SOFTUNE Workbench is not in the debug session, the specified source file is displayed in the edit window.

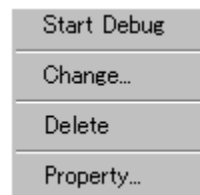
- Property

File information is displayed. (See Section "4.3.9 Property").

■ Short-Cut Menu (Click the Right Button of the Mouse on Setup Name)

Figure 3.4-17 shows a short-cut menu.

Figure 3.4-17 Short-Cut Menu on a Setup Name



- Start /End Debug

During debugging session, the confirmation dialog 'Is the current session ended to start debugging?' is displayed. Select Yes.

The current debugging is ended, and then debugging is started based on the specified setup information.

During non-debugging session, debugging is started without displaying the confirmation dialog.

Debugging can be started only in the active project.

In debug session, only the valid setup changes from [Start Debug] to [End Debug]. When [End Debug] is selected, debugging is ended.

- Change

The setup wizard for changing the setup setting is started (see Section "4.7.2.4 Setup Wizard").

- Delete

Debugger setup information is deleted from the project.

Valid setup information cannot be deleted. To delete, change the valid setup temporarily.

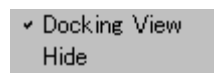
- Property

Setup information is displayed (see Section "4.3.9 Property").

■ Short-Cut Menu (Click the Right Button of the Mouse on a Space in the Project Window)

Figure 3.4-18 shows a short-cut menu.

Figure 3.4-18 Short-Cut Menu on Space in the Project Window



- Docking View

The project window is docked with the frame in the check status.

- Hide

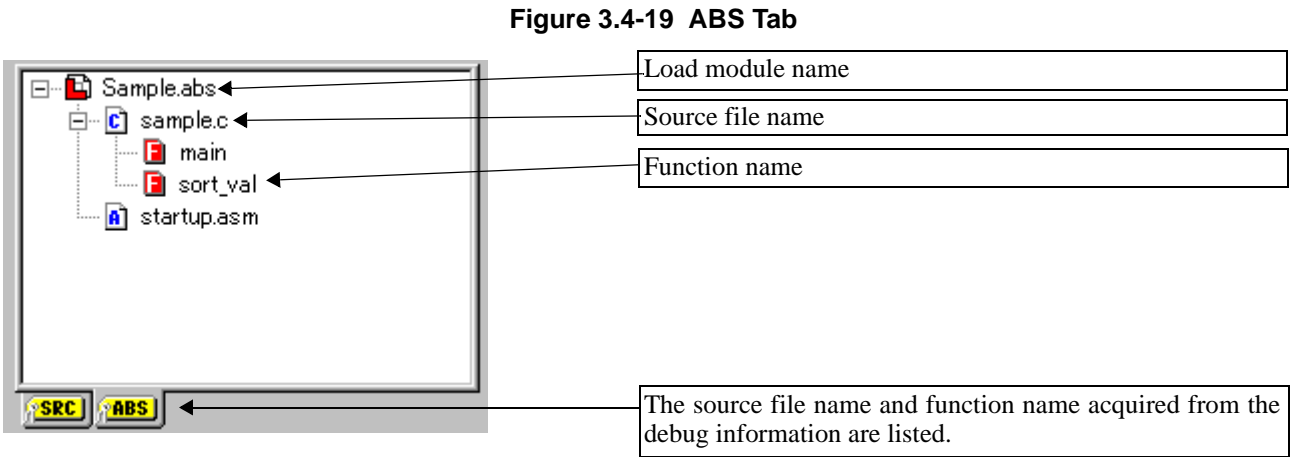
The project window enters the nondisplay status.

3.4.2 ABS Tab

The load module name, source file name, and function name are used as the information acquired from the debug information, and are displayed in the tree view format.

■ ABS Tab

Figure 3.4-19 shows examples of displayed contents of the ABS tab.



■ Function

Display the following information from the debug information.

- Load module name
- Source file name

Double-click the source file name to open the source window.

- Function name

Double-click the function name to jump to the source window of the function definition position.





- Coverage rate

Indicates the coverage rates of load module, source file, and function. Update of the ABS tab is required to display the latest coverage rate.

The coverage rate is displayed as "(--%)" when all areas are out of range of the coverage measurement. If a part of the area falls in the outside of the coverage measurement range, an asterisk "*" is attached to the end of the coverage rate.

This function only becomes valid while MB2141 emulator debugger.

■ ABS Tab Bitmap Image List

-  Load module name
-  C source file
-  Assembler source file
-  Function name

■ Short-Cut Menus (Click the Right Button of the Mouse on a Load Module Name)

Figure 3.4-20 shows a short-cut menu.

Figure 3.4-20 Short-Cut Menu on a Load Module Name



- Open
Cannot be selected.
- Order by name
Sort items in the tree in alphabetical order.
- Order by address
Sort functions in the tree in address order, and sort other items in the tree in alphabetical order.
- Order by coverage rate
Sort items in the tree in ascending order.
It is only valid while MB2141 emulator debugger.
- Set coverage
Opens the coverage setting dialog.
It is only valid while MB2141 emulator debugger.
- Display coverage rate
Switches on/off the display of the coverage rate. When Display coverage rate is turned on, the coverage rate of each item is displayed. It is only valid while MB2141 emulator debugger.
- Refresh
Updates the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141 emulator debugger.
- Property
Information on the load module file is displayed (See Section "4.3.9 Property").

■ Short-Cut Menus (Click the Right Button of the Mouse on a Source File Name)

Figure 3.4-21 shows a short-cut menu.

Figure 3.4-21 Short-Cut Menu on a Source File Name



- Open
Opens the selected source file in the source window.
- Order by name
Sort items in the tree in alphabetical order.
- Order by address
Sort functions in the tree in address order, and sort other items in the tree in alphabetical order.
- Order by coverage rate
Sort items in the tree in ascending order.
It is only valid while MB2141 emulator debugger.
- Set coverage
Opens the coverage setting dialog.
It is only valid while MB2141 emulator debugger.
- Display coverage rate
Switches on/off the display of the coverage rate. When Display coverage rate is turned on, the coverage rate of each item is displayed. It is only valid while MB2141 emulator debugger.
- Refresh
Updates the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141 emulator debugger.
- Property
Display the properties of the source file.

■ Short-Cut Menus (Click the Right Button of the Mouse on a Function Name)

Figure 3.4-22 shows a short-cut menu.

Figure 3.4-22 Short-Cut Menu on a Function Name

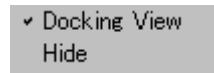


- Jump
Opens the selected function in the source window.
- Order by name
Sort items in the tree in alphabetical order.
- Order by address
Sort functions in the tree in address order.
- Order by coverage rate
Sort items in the tree in ascending order.
It is only valid while MB2141 emulator debugger.
- Break
Opens a break dialog to set a break point at the selected function.
- Set coverage
Opens the coverage setting dialog.
It is only valid while MB2141 emulator debugger.
- Display coverage rate
Switches on/off the display coverage rate. When Display coverage rate is turned on, the coverage rate of each items is displayed. It is only valid while MB2141 emulator debugger.
- Refresh
Update the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141 emulator debugger.
- Property
Display the properties of the symbol information of the selected function.

■ Short-Cut Menu (Click the Right Button of the Mouse on Space in the Project Window)

Figure 3.4-23 shows a short-cut menu.

Figure 3.4-23 Short-Cut Menu on Space in the Project Window



- Docking View

The project window is docked with the frame in the check status.

- Hide

The project window enters the nondisplay status.

3.5 Output Window

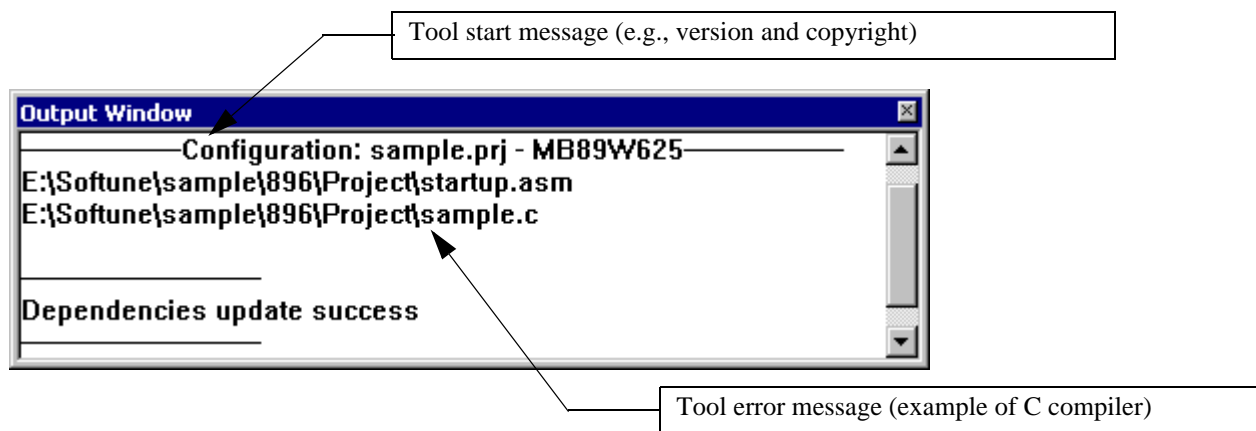
The output window displays compiler version information and error messages, etc., when make or build is executed.

■ Output Window

Figure 3.5-1 shows an example of the output window.

As shown in this example, when the left button of the mouse is double-clicked on the line to which an error message was output, control automatically jumps to the source line where the error occurred.

Figure 3.5-1 Example of Output Window



■ Short-Cut Menu

There are seven menus below:

- | | |
|--------------------|---|
| - Copy | Copies the selected character string to the clipboard. |
| - Clear | Clears all the items displayed in the output window. |
| - Docking View | Sets or resets the docking view. |
| - Nondisplay | Places the output window in the nondisplay status. |
| - Set Font | Selects a display font in the output window. |
| - Keyword Coloring | Whether or not to highlight the error number of the error message is set. |
| - Logging | The result of output to the output window is stored in a file. |

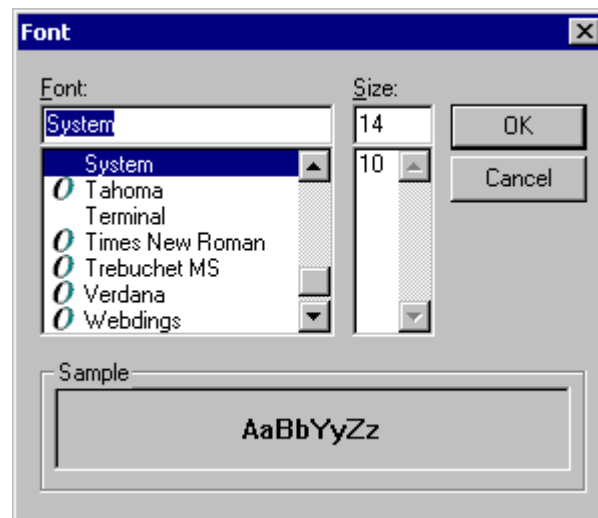
■ Operation

- Copy
 1. Select the character string you want to copy.
 2. Click the right button of the mouse in the output window to display the short-cut menu.
 3. Select [Copy] from the short-cut menu.

See Section "4.3.2 Cut, Copy, Paste, Delete".
- Clear
 1. Click the right button of the mouse in the output window to display the short-cut menu.

2. Select [Clear] from the short-cut menu.
- Docking View
 1. Click the right button of the mouse in the output window to display the short-cut menu.
 2. Select [Docking View] from the short-cut menu.
 - The output window is docked with the frame in the check status.
- Display
 1. Click the right button of the mouse in the output window to display the short-cut menu.
 2. Select [Nondisplay] from the short-cut menu.
 - The output window enters the nondisplay status.
- Set Font
 1. Click the right button of the mouse in the output window to display the short-cut menu.
 2. Select [Set Font] from the short-cut menu.
 - The font selection dialog (Figure 3.5-2) opens.
 3. Select [Font] and [Size], then click the [OK] button.

Figure 3.5-2 Font Setting Dialog



- Highlight Keyword
 1. Click the right button of the mouse in the output window to display the short-cut menu.
 2. Select [Keyword Coloring].
 - In the check state, the keyword (error number) of the error message is displayed in red.
- Logging
 1. Click the right button of the mouse in the output window to display the short-cut menu.
 2. Select [Logging]-[Start] and then specify the file name of the record target file.
 - The result of output to the Output window is recorded in the specified file.
 3. To suspend logging, select [Logging]-[Record] and uncheck the checkbox.
 - One selected, logging is restarted. When the checkbox is checked, the file is recorded.
 4. To stop logging, select [Logging]-[Termination].

3.6 Edit Window (Standard Editor)

The edit window is used to display and edit a source file and document file, etc.

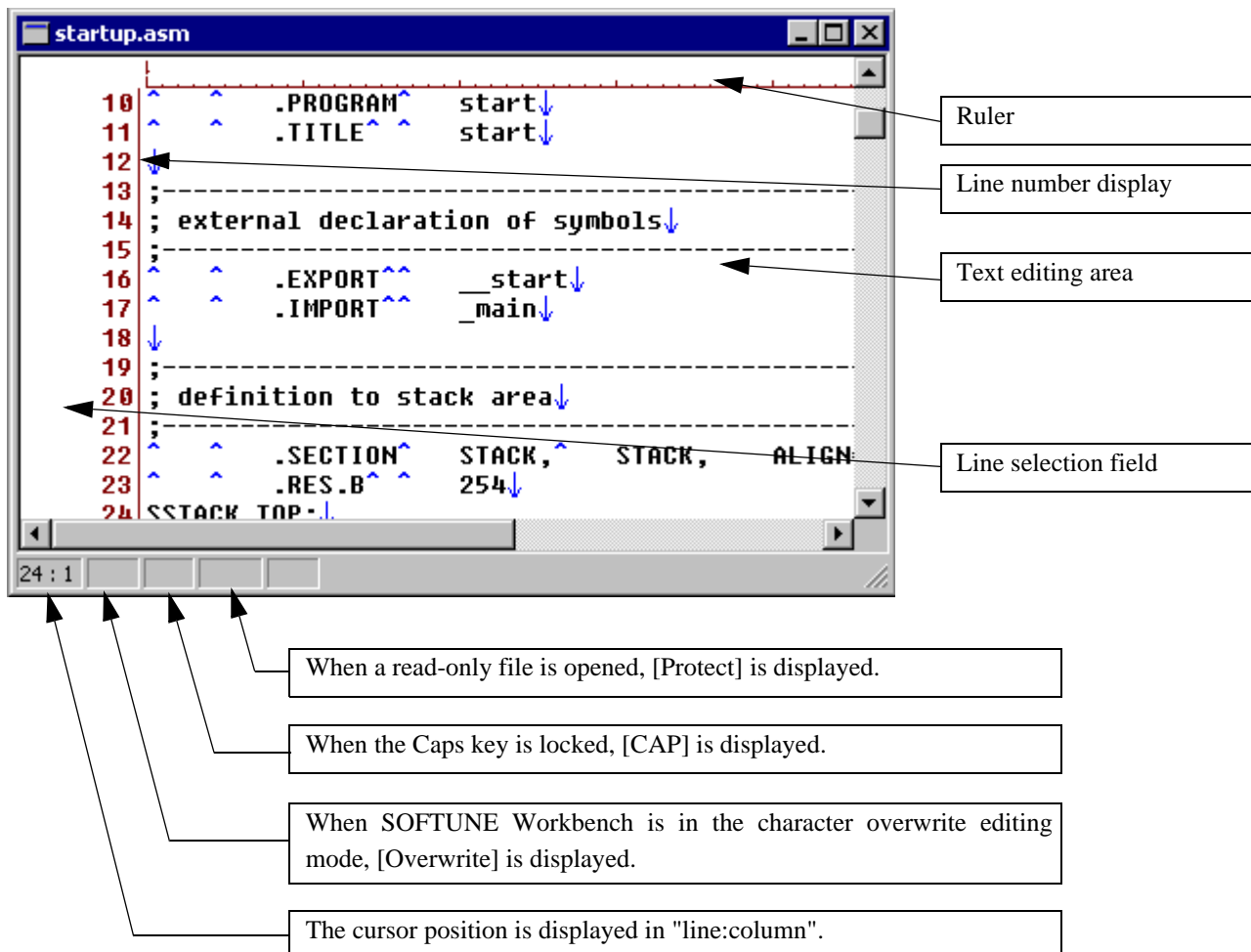
■ Edit Window

An example of the edit window docked with the Main Window is given in Figure 3.6-1.

As shown in this example, the edit window docked with the Main Window can be displayed only when the standard editor built into SOFTUNE Workbench is in use.

SOFTUNE Workbench can register and use any external editor. In this case, however, the edit window is not treated as a SOFTUNE Workbench screen.

Figure 3.6-1 Example of Edit Window



■ Short-Cut Menu

- | | |
|-----------------|---|
| - Source Window | Opens the Source Window. |
| - Undo | Cancels the last editing. |
| - Redo | Redoes the canceled editing. |
| - Cut | Moves the selected character string to the clipboard. |

- Copy Copies the selected character string to the clipboard.
 - Paste Pastes the character string in the clipboard to the cursor position.
 - Delete Deletes the selected character string.
 - All select Selects all characters in the buffer.
 - Print Prints the file.
 - Add to Project Adds the file being edited to the active project.
 - Find Opens the character string find dialog.
 - Replace Opens the character string replacement dialog.
 - Find in Files See section "4.3.5 Find in Files".
 - Jump Opens the jump line number specification dialog.
 - Bookmark See Section "4.3.7 Bookmark".
 - Set Standard Editor See Section "3.6.1 Setting Standard Editor".
 - Status Bar Switches status bar display for each window.
 - Property Display file information (properties).
-
- Source Window
 1. Click the right button of the mouse to display the short-cut menu, then select [Source Window] from the menu.
 - When SOFTUNE Workbench is in the debug session, the Source Window is displayed. (See Section "3.7 Source Window".)
 - When SOFTUNE Workbench is not in the debug session, this menu cannot be selected
-
- Undo
 1. Click the right button of the mouse to display the short-cut menu, then select [Undo] from the menu.
 - SOFTUNE Workbench cancels the last editing and undoes the status before the editing. (See Section "4.3.1 Undo, Redo".)
-
- Redo
 1. Click the right button of the mouse to display the short-cut menu, then select [Redo] from the menu.
 - SOFTUNE Workbench cancels the last editing. (See Section "4.3.1 Undo, Redo".)
-
- Cut
 1. Select the character string you want to cut.
 2. Click the right button of the mouse to display the short-cut menu, then select [Move] from the menu.
 - SOFTUNE Workbench deletes the selected character string from the edit window and moves it to the clipboard. (See Section "4.3.2 Cut, Copy, Paste, Delete".)
-
- Copy
 1. Select the character string you want to copy.
 2. Click the right button of the mouse to display the short-cut menu, then select [Copy] from the menu.
 - SOFTUNE Workbench copies the selected character string to the clipboard. (See Section "4.3.2 Cut, Copy, Paste, Delete".)
-
- Paste
 1. Move the cursor to the position into which the character string is to be inserted.
 2. Click the right button of the mouse to display the short-cut menu, then select [Paste] from the menu.

- SOFTUNE Workbench inserts the character string in the clipboard into the cursor position. (See Section "4.3.2 Cut, Copy, Paste, Delete".)
- Delete
 1. Select the character string you want to delete.
 2. Click the right button of the mouse to display the short-cut menu, then select [Delete] from the menu.
 - SOFTUNE Workbench deletes the selected character string. (See Section "4.3.2 Cut, Copy, Paste, Delete".)
- All select
 1. Click the right button of the mouse to display the short-cut menu, then select [All select] from the menu.
 - SOFTUNE Workbench selects all the texts in the edit window and displays them in reverse video. (See Section "4.3.3 All Select".)
- Print
 1. Click the right button of the mouse to display the short-cut menu, then select [Print] from the menu.
 - The print dialog opens. (See Section "4.2.9 Print".)
- Add to Project
 1. Click the right button of the mouse to display the short-cut menu, then select [Add to Project] from the menu.
 - SOFTUNE Workbench adds the file being edited to the project.
- Find/Replace
 1. Click the right button of the mouse to display the short-cut menu, then select [Find] or [Replace] from the menu.
 - SOFTUNE Workbench sets the clipboard character string in [Find What] and opens the find or replace dialog. (See Section "4.3.4 Find/Replace".)
- Find in files
 1. Click the right button of the mouse to display the short-cut menu, then select [Find in files] from the menu.
 - Opens the [Find in Files] dialog. (See Section "4.3.5 Find in Files".)
- Jump
 1. Click the right button of the mouse to display the short-cut menu, then select [Jump] from the menu.
 - The dialog for specifying the jump destination line opens. (See Section "4.3.6 Jump".)
 2. Set the jump destination line number.
 3. Click the [OK] button.
- Status Bar

Status Bar switches status bar display and non-display for each Edit Window.

 1. Click the right button of the mouse to display the short-cut menu, then select [Status Bar] from the menu.
 - When the status bar has been already displayed, SOFTUNE Workbench switches status bar display to nondisplay and deletes the check mark from the menu.
 - When the status bar is not displayed, SOFTUNE Workbench displays the status bar and adds a check mark to the left end of the menu.
- Property
 1. Click the right button of the mouse to display the short-cut menu, then select [Property] from the list.
 - SOFTUNE Workbench displays file information. (See Section "4.3.9 Property".)

3.6.1 Setting Standard Editor

This section explains how to customize the standard editor.

■ Setting the Standard Editor

Some standard editor functions can be customized.

- Items that can be customized
 - Display function
 - Line feed mark display
 - Tag code mark display
 - EOF code mark display
 - Line number display
 - Zenkaku spc display
 - Ruler display
 - Automatic indent function
 - Colored display of C keyword
 - Colored display of ASM keyword
 - Colored display of user keyword
 - Colored display of comment statement
 - Font
 - Tab count
 - Display color
 - Bookmark
 - Error line
 - C keyword
 - Assembly keyword
 - User keyword
 - Comment statement
 - Line Feed
 - Tab
 - Zenkaku Spc
 - Keyword to be highlighted

■ Customization Procedure (Display Function Selection)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Select the function item you want to set or reset from the "Item" list, then click to the left of the item.
3. When not setting other items, click the [OK] button.
When selecting a color, set all the related items, then click the [OK] button.

■ Customization Procedure (Display Color Selection)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Click the color button of [Bookmark], [Error], [C], or [Assembly]. From [View2] tab, you can set a [Comment], [Ret], [Tab], [User Keyword], and [Zenkaku Spc].
The color dialog opens.
3. Select a color from the dialog, then click the [OK] button.
The color of the button changes to the selected color.
4. When not setting other items, click the [OK] button.
When selecting a font, set all the related items, then click the [OK] button.

■ Customization Procedure (Font Selection)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Click the [Font] button.
The font selection dialog opens.
3. Select a font and size, then click the [OK] button.
4. When not setting other items, click the [OK] button.
When specifying a tab count, set all the related items, then click the [OK] button.

■ Customization Procedure (Keyword Addition)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Click [View2] tab to choose the keyword type to be changed from the combo box.
Click [Detail...] button.
The keyword dialog shown in Figure 3.6-4 opens.
3. Enter the keyword to be highlighted in [New Item] field. Click [Add] button.
4. Click the [OK] button.
The keyword files is changed.
5. When not setting other items, click the [OK] button.
When selecting a display function, set all the related items, then click the [OK] button.

■ Customization Procedure (Tab Count)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Modify the value displayed in the [Tab Count] field.
3. When not setting other items, click the [OK] button.
When selecting a display function, set all the related items, then click the [OK] button.

■ Customization Procedure (Deletion of Keyword)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Click [View2] tab to choose the keyword type to be changed from the combo box.
Click [Detail...] button.
The keyword dialog shown in Figure 3.6-4 opens.
3. Choose the keyword to be deleted from the [Keyword List]. Click [Delete] button.
4. Click the [OK] button.
The keyword files is changed.
5. When not setting other items, click the [OK] button.
When selecting a display function, set all the related items, then click the [OK] button.

■ Customization Procedure (Initialization of Keyword)

1. Click the right button of the mouse in the edit window to display the short-cut menu, then select [Customize...] from the menu.
The standard editor dialog shown in Figure 3.6-2 opens.
2. Click [View2] tab to choose the keyword type to be changed from the combo box.
Click [Detail...] button.
The keyword dialog shown in Figure 3.6-4 opens.
3. Click [Reset] button.
4. Click the [OK] button.
The keyword files is changed.
5. When not setting other items, click the [OK] button.
When selecting a display function, set all the related items, then click the [OK] button.

Figure 3.6-2 Standard Editor Dialog

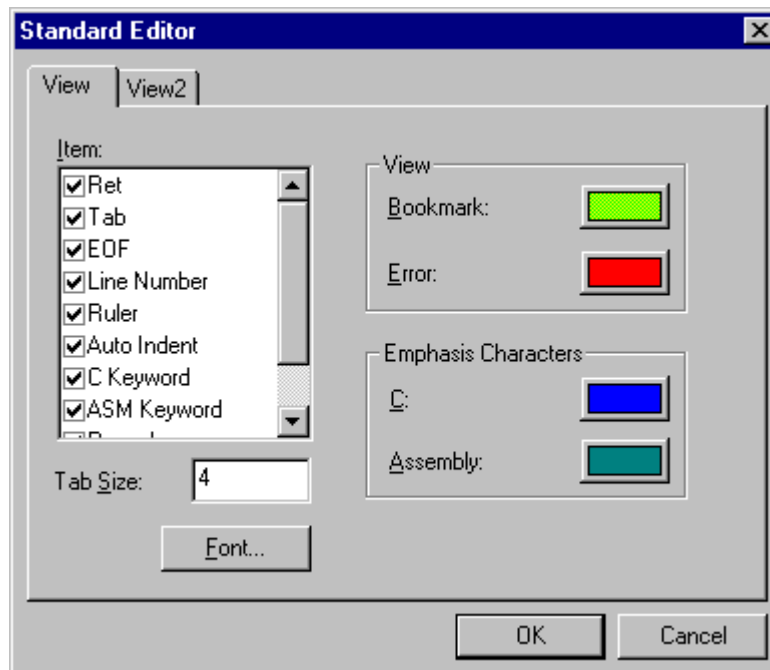


Figure 3.6-3 Standard Editor Dialog (View2)

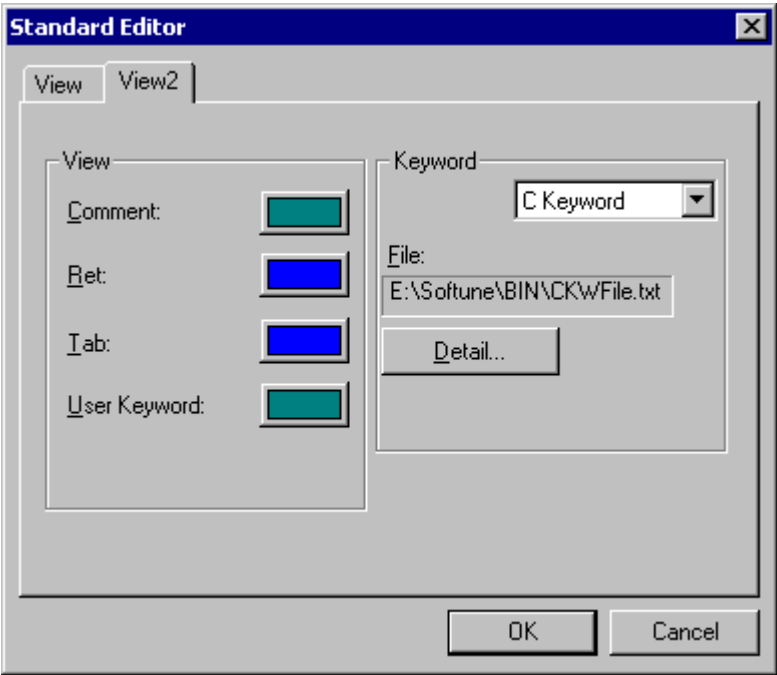
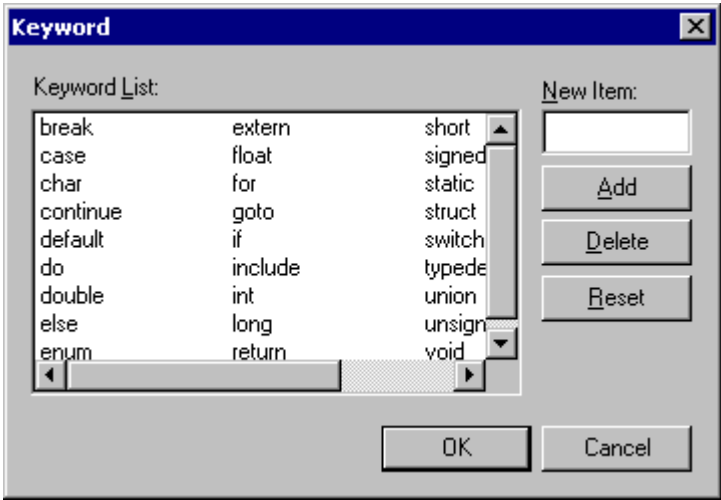


Figure 3.6-4 Keyword Dialog



3.7 Source Window

The source window displays the source program at debugging.

■ Source Window

Figure 3.7-1 shows an example of the source window.

■ Function

- Break point setting/releasing

A break point can be set or released by left-clicking '==>O' of 'O' at the left edge of the source window.

- Instruction execution to specified line (address)

An instruction can be executed at the specified line (address) by left-clicking of '==>O' of '==>' at the left edge of the source window. Step execution (step/into) is also enabled by clicking at the PC location.

- Pop-up display of variable value

By placing cursor of mouse on the name of variable, that variable value can be viewed as pop-up.

- Bookmark setup function

By setting a bookmark, it enables to view the specified line using search bar and menu or key operation. Please refer to "4.3.7 Bookmark" for details.

- Drag and Drop

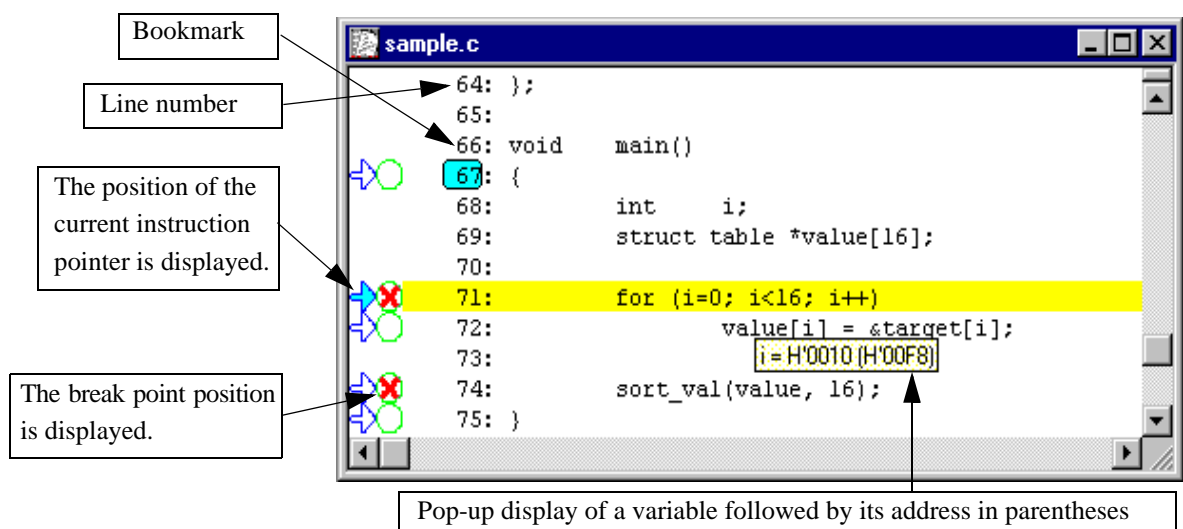
It is possible to drag a character string selected at source window and drop it to memory, disassemble, and watch windows.

Please refer to description about "drag and drop" at each window for operation done after dropping character string.

- Display coverage

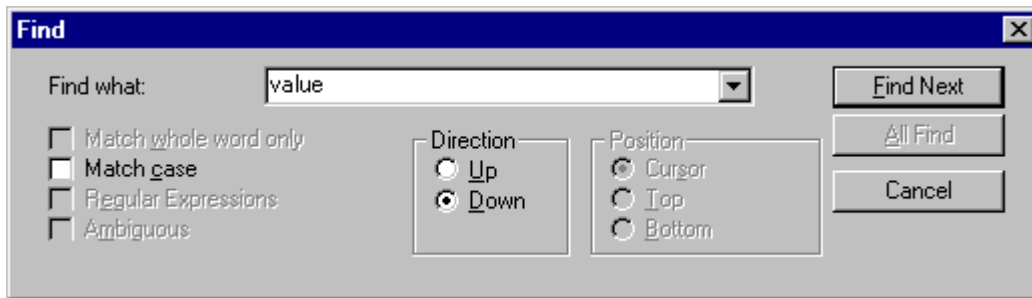
Coverage can be displayed on a source line basis from the short-cut menu [Display coverage]. In Mix Display, coverage is displayed on a machine instruction basis. Update of the source window is required to display the latest coverage. It is only valid while MB2141 emulator debugger.

Figure 3.7-1 Example of Source Window



■ Short-Cut Menu

- Edit: Opens the edit window.
 - Update: Updates the contents of the specified window. When Display coverage is turned on, the display is updated with the latest coverage. It is only valid while MB2141 emulator debugger.
 - Inline Assemble: See Section "4.4.3 Assembly".
 - Find: Opens the character string search dialog (Figure 3.7-2).
 - Jump: Displays jump dialog.
 - Go to Current: Displays current PC location.
 - Break point Set/Reset: This function is to set or to cancel the break point in the line pointed by cursor.
 - Break point Enable/Disable: To make break point in the line pointed by cursor enable or disable.
 - Break: See Section "4.6.4 Break Point".
 - Watch: Additionally sets watch point
 - Stack: See Section "4.6.7 Stack".
 - Set coverage: Please refer to Section "4.4.12 Coverage". It is only valid while MB2141 emulator debugger.
 - Property: Displays source window properties.
 - Display coverage: Switches on/off the display of the coverage. When Display coverage is turned on, the executed lines are displayed in green and the unexecuted lines are in brown. It is only valid while MB2141 emulator debugger.
 - Mix Display: Switches whether to display or not disassemble list.
 - Activate when stop: Specify whether the window should set to active or not when the program is stopped.
 - Close: Closes source window.
-
- Edit
 - Opens the edit window to modify the current source file.
 - (See Section "3.6 Edit Window (Standard Editor)".)
 - Find
 - Searches the text in the current source file. Perform this operation in the following procedure:
 1. Click the right button of the mouse in the source window to display the short-cut menu, then select [Find] from the menu.
 - The search dialog shown in Figure 3.7-2 opens.
 2. Set the character string you want to search.
 3. Set a search direction.
 4. If necessary, display a check mark to [Match case].
 5. Click the [Find Next] button.
 - The found character string is displayed in reverse video.
 6. To terminate search, click the [Cancel] button.

Figure 3.7-2 Find Dialog in Source Window

- Jump

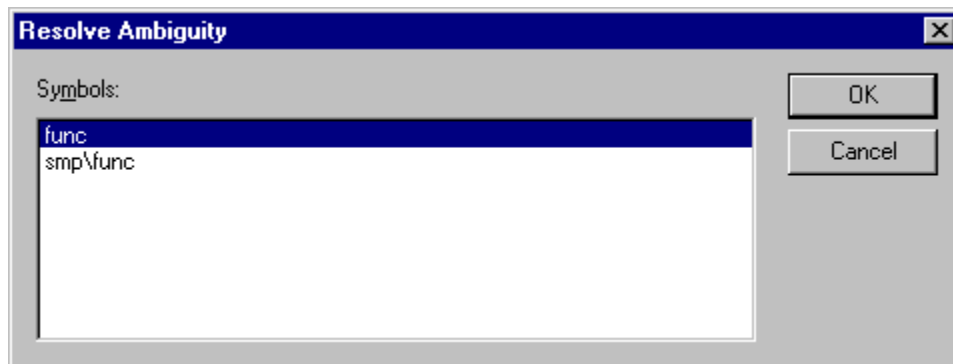
Specifies the position to display in the source window. Perform this operation in the following procedure:

1. Click the right button of the mouse in the source window to display the short-cut menu, then select [Jump] from the menu.
(See Section "4.3.6 Jump".)
2. Specify a specification format.
3. Select a display position.
4. Click the [OK] button.

- Move to Defined Line

Move to the defined line of function. Perform this operation in the following procedure:

1. Select the function name in the source window or move the cursor on the function name.
2. Click the right button of the mouse to display the short-cut menu, then select [Move to Defined Line] from the menu.
3. If the same function names exist, the [Resolve Ambiguity] dialog is displayed, then select the function name and click the [OK] button.

Figure 3.7-3 Resolve Ambiguity Dialog

- Break Point Set/Reset

After moving cursor to the line where you want to set or to cancel a break point, click the right button to display short-cut menu. Then, please select [Break Point Set/Reset].

- Break Point Enable/Disable

After moving cursor to the line where break point is set to be enable or disable, click right button to display the short-cut menu. Then, please select [Break Point Enable/Disable]. This enables to switch the break point, which is pointed by cursor, enable or disable.

- Watch

Specifies the variable to watch and opens the watch window. Perform this operation in the following procedure:

1. Click the right button of the mouse in the source window to display the short-cut menu, then select [Watch] from the menu.
(See Section "4.4.7 Watch".)
2. Specify a variable name.
3. Select a [mode] as required.
4. Click the [OK] button.

- Property

Click the right button of the mouse to display the short-cut menu, then select [Property] from the menu. File information is displayed (see Section "4.3.9 Property").

- Mix Display

1. Click the right button of the mouse in the source window to display the short-cut menu.
2. Check [Mix Display].
 - When a check mark is displayed to the left of [Mix Display], mix display is already set.
 - When a check mark is not displayed to the left of [Mix Display], mix display is not set.
3. Select [Mix Display].
 - When it is already set, mix display is reset and the check mark is deleted.
 - When it is not set, mix display is set and a check mark is displayed.

- Display coverage

1. Click the right button of the mouse in the source window to display the short-cut menu.
2. Check [Display coverage].

A check mark attached to the left side of [Display coverage] indicates that display coverage has already been turned on.

If a check mark is not attached to the left side of [Display coverage], display coverage is turned off.
3. Select [Display coverage].
 - Display coverage is turned off, and the check mark is disappeared when it has already been turned on.
 - Display coverage is turned on, and a check mark is attached when it has been turned off.

Note:

The coverage display in the source window is not automatically updated. You need to update the display with the latest information using the short-cut menu [Update] or by selecting [Update window] or [Update all windows] from the main menu.

- Activate when stop

Specify whether the source window should set to active or not when the program is stopped. However, if a source window is displayed as new, it should always be activated. Perform this operation in the following procedure:

1. Click the right button of the mouse in the source window to display short-cut menu.
2. Check [Activate when stop].

If a check mark is placed on the left side of [Activate when stop], source file, which corresponds to break address while halting program execution, should be activated. On the other hand, if the source file is not displayed, open a source file that corresponds to break address regardless of specification.

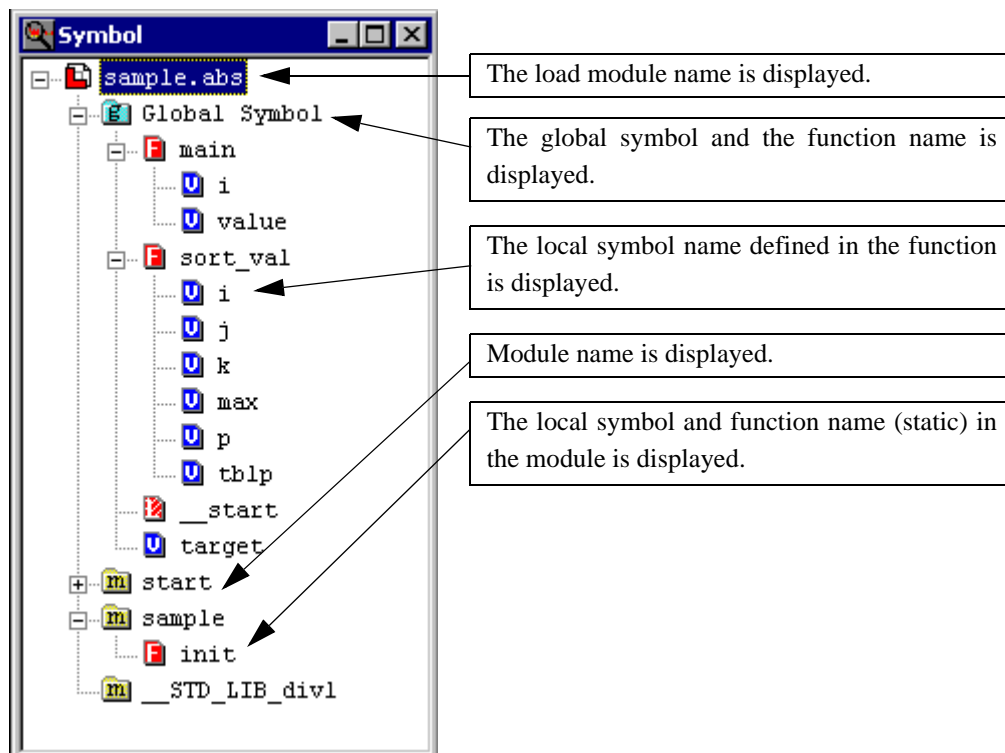
3.8 Symbol Window

The Symbol Window displays the target file name of the current project and the symbol names (e.g., variable name and function name) being used by the file in the tree view format.

■ Symbol Window

Figure 3.8-1 shows an example of the Symbol Window.

Figure 3.8-1 Example of Symbol Window



■ Short-Cut Menu

Jump	Opens the Source Window of the specified function.
Watch	Sets specified symbol as watch point.
Break	See Section "4.6.4 Break Point".
Property	Displays symbol properties.
Mangle name	Specify whether to display a symbol name as a mangled name.
Refresh	The latest symbol information is displayed.
Close	Closes Symbol Window.

- Jump

1. Click the right button of the mouse on the function name to display the short-cut menu, then select [Jump] from the menu.

The source window in which the function is defined opens.

- Watch
 1. Click the right button of the mouse on the variable or function name to display the short-cut menu, then select [Watch] from the menu. The variable or function is added to the watch and the Watch Window is displayed.
- Property
 1. Click the right button of the mouse to display the short-cut menu, then select [Property] from the menu. File information is displayed (see Section "4.3.9 Property").
- Mangle name
 1. Click the right button of the mouse to display the short-cut menu and select [Mangled name]. Switch between display and non-display of a symbol name as a mangled name.
- Refresh
 1. Click the right button of the mouse to display the short-cut menu and select [Refresh]. The latest symbol information is displayed.

3.9 Disassemble Window

This window is displayed only when SOFTUNE Workbench is in the debug session. The disassemble window displays the result obtained by executing disassembling at the specified address.

■ Disassemble Window

Figure 3.9-1 shows an example of the disassemble window.

■ Function

- Break point setting/releasing

A break point can be set or released by left-clicking '==> ○' of '○' at the left edge of the disassemble window.

- Instruction execution at specified line (address)

An instruction can be executed at the specified line (address) by left-clicking of '==>○' of '==>' at the left edge of the disassemble window. Step execution (step/into) is also enabled by clicking the left mouse button at the PC location.

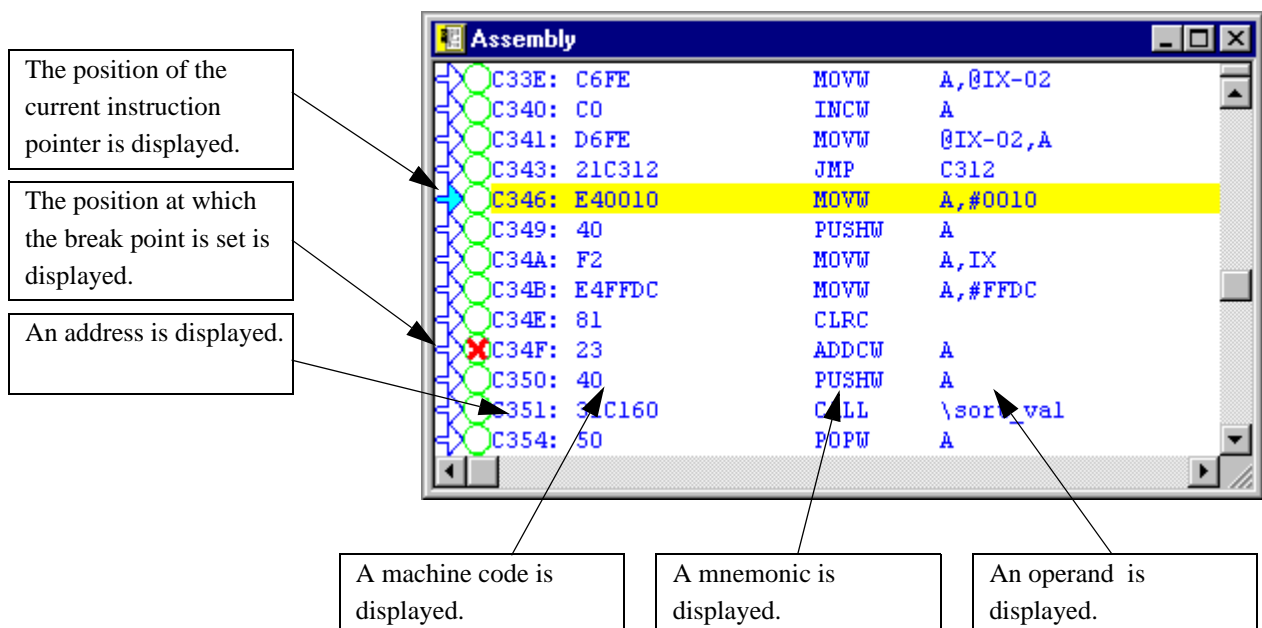
- Drag and Drop

By dropping function name, label, or address from source window to disassemble window, display will be jumped to a location where an address of dropped character string is shown.

- Display coverage

Coverage can be displayed on a machine instruction basis from the short-cut menu [Display coverage]. It is only valid while MB2141 emulator debugger.

Figure 3.9-1 Disassemble Window



■ Short-Cut Menu

Inline Assemble:	See Section "4.4.3 Assembly".
Jump:	Displays jump dialog.
Go to Current:	Displays current PC location.
Break Point Set/Reset:	This function is to set or to cancel the break point in the instruction pointed by cursor.
Break Point Enable/Disable:	To make break point in the instruction pointed by cursor enable or disable.
Break:	See Section "4.6.4 Break Point".
Watch:	See Section "4.4.7 Watch".
Stack:	See Section "4.6.7 Stack".
- Set coverage:	Please refer to Section "4.4.12 Coverage". It is only valid while MB2141 emulator debugger.
- Display coverage:	Switches on/off the display of the coverage. When Display coverage is turned on, the executed lines are displayed in green and the unexecuted lines are in brown. It is only valid while MB2141 emulator debugger.break point
Activate when stop:	Specify whether the window should set to active or not when the program is stopped.
Close:	Closes disassemble window.

- Jump

Specifies the position to display in the disassemble window. Perform this operation in the following procedure:

1. Click the right button of the mouse in the disassemble window to display the short-cut menu, then select [Jump] from the menu.
The jump dialog opens. (See Section "4.3.6 Jump").
2. Select a specification format.
3. Specify a cursor display position.
4. Click the [OK] button.

- Display coverage

1. Click the right button of the mouse in the disassemble window to display the short-cut menu.
2. Check [Display coverage].

A check mark attached to the left side of [Display coverage] indicates that display coverage has already been turned on.

If a check mark is not attached to the left side of [Display coverage], display coverage is turned off.

3. Select [Display coverage].

- Display coverage is turned off, and the check mark is disappeared when it has already been turned on.
- Display coverage is turned on, and a check mark is attached when it has been turned off.

- Break Point Set/Reset

After moving cursor to the instruction where you want to set or to cancel a break point, click the right button to display short-cut menu. Then, please select [Break Point Set/Reset].

- Break Point Enable/Disable

After moving cursor to the instruction where break point is set to be enable or disable, click right button to display the short-cut menu. Then, please select [Break Point Enable/Disable]. This enables to switch the break point, which is pointed by cursor, enable or disable.

- Activate when stop

Specify whether the disassemble window should set to active or not when the program is stopped. However, if a disassemble window is displayed as new, it should always be activated. Perform this operation in the following procedure:

1. Click the right button of the mouse in the disassemble window to display short-cut menu.
2. Please make sure [Activate when stop].

3.10 Register Window

This window is displayed only when SOFTUNE Workbench is in the debug session. The Register Window is used to display and modify MCU register values during debugging.

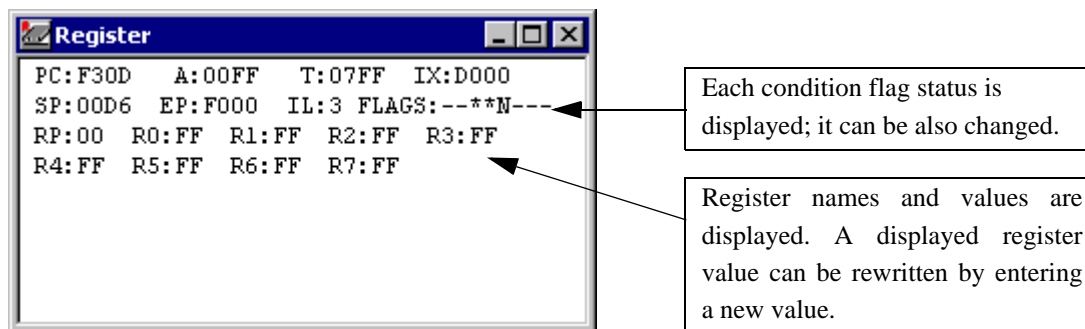
■ Register Window

Figure 3.10-1 shows an example of the Register Window. When debugging, the Register Window displays the register values. Register values can be rewritten directly. To rewrite values using expressions and symbols, etc., select [Edit] from the short-cut menu for the Register Window.

Arrange registers according to the size of the Register Window.

To select the registers to be displayed, select [Setup] from the short-cut menu for the Register Window.

Figure 3.10-1 Example of Register Window



■ Short-Cut Menu

- Setup Setup See Section "3.10.1 Setting Register Display".
- Edit Displays Register Edit Dialog.
- Close Closes Register Window.

- Edit

Edits register values. Beside this operation, the numerical values displayed in the Register Window can be directly edited. (See Section "4.4.4 Register".)

1. Click the right button of the mouse in the Register Window to display the short-cut menu, then select [Edit] from the menu.
The register edit dialog opens.
2. Select a register name.
3. Enter the value you want to set.
4. Click the [OK] button.

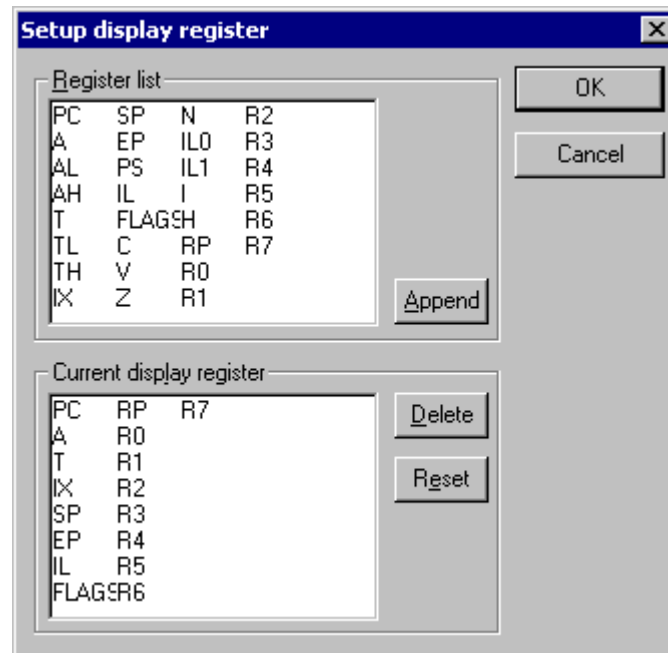
3.10.1 Setting Register Display

This section explains how to set register display.

■ Setting Register Display

Set the registers to be displayed in the Register Window in the following procedure:

Figure 3.10-2 Display Register Setting Dialog



■ Adding Registers

Add the register you want to display in the following procedure:

1. Click the right button of the mouse in the Register Window to display the short-cut menu, then select [Setup] from the menu.
The display setting dialog shown in Figure 3.10-2 opens.
2. Select the register you want to display from [Register List] (upper part of display register setting dialog), then click the [Append] button.
The selected register is set in [Current Display Register] (lower part).
3. Set all the registers you want to display by repeating the above operation.
4. Click the [OK] button.

■ Deleting Registers

Select the register you want to delete in the following procedure:

1. Click the right button of the mouse in the Register Window to display the short-cut menu, then select [Setup] from the menu.
The display setting dialog shown in Figure 3.10-2 opens.
2. Select the register you want to delete from [Current Display Register] (lower part of display register setting dialog), then click the [Delete] button.

The selected register is deleted from [Current Display Register] (lower part).

3. Set all the registers you want to delete by repeating the above operation.
4. Click the [OK] button.

■ Restoring Registers to the Initial Status

Restore the registers in [Current Display Registers] to the initial status (status when SOFTUNE Workbench was installed) in the following procedure:

1. Click the right button of the mouse in the Register Window to display the short-cut menu, then select [Setup] from the menu.

The display setting dialog shown in Figure 3.10-2 opens.

2. Click the [Reset] button.

The registers in [Current Display Registers] (lower part of display register setting dialog) are restored to the initial status.

3. Click the [OK] button.

3.11 Memory Window

This window opens only when SOFTUNE Workbench is in the debug session. The memory window is used to display and modify memory data at the specified address.

■ Memory Window

Figure 3.11-1 shows an example of the memory window. When debugging, memory values can be displayed. The register values can be directly rewritten.

To modify by specifying the expression and symbol in the memory values, select [Edit] from the short-cut menu for the memory window.

■ Function

- Display the change

The location, where was changed from the previous contents, will be shown in red while program execution is stopped or monitoring. This function is only valid in a range where currently displayed by memory window.

- Display with 16 bytes fixed in size

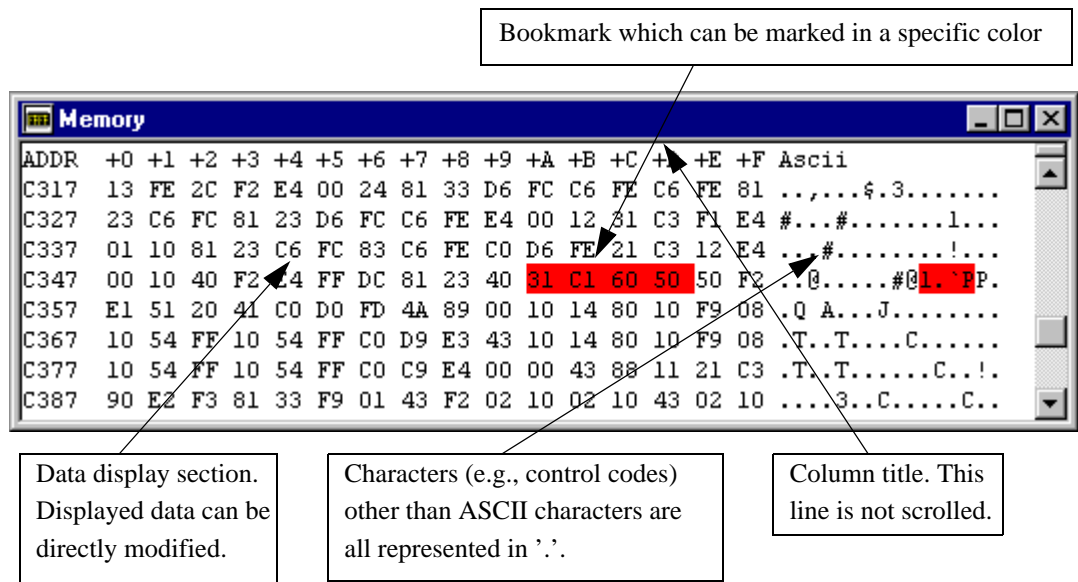
The number of bytes at one line can be chosen: [Auto] which is corresponding to the size of window or [4byte], [8byte], [16byte], [32byte] and [64byte] which the size of bytes are fixed. In order to set up the number of bytes, use [Setup] at short-cut menu.

- Bookmark setup function

Bookmark can be set by coloring the specified address range. By setting a bookmark, it enables to view the specified line using search bar and menu or key operation. Please refer to "4.3.7 Bookmark" for details.

- Drag and Drop

By dropping variable name, function name, label or address from source window to memory window, the address of dropped character string is shown and address range of the symbol will be chosen.

Figure 3.11-1 Example of Memory Window

■ Short-Cut Menu

- | | |
|--------------------|---|
| - Compare: | See Section "4.4.5 Memory". |
| - Find: | See Section "4.4.5 Memory". |
| - Special: | See Section "4.4.5 Memory". |
| - Inline Assemble: | See Section "4.4.3 Assembly". |
| - Jump: | See Section "4.3.6 Jump". |
| - Bookmark: | See Section "4.4.5 Memory". |
| - Edit: | See Section "4.4.5 Memory". |
| - Setup: | See Section "4.4.5 Memory". |
| - Break: | See Section "4.6.4 Break Point". |
| - Event: | See Section "4.6.5 Event". |
| - Watch: | See Section "4.4.7 Watch". |
| - Monitoring: | Switches whether to enable or disable monitoring. |
| - Close: | Closes the memory window. |

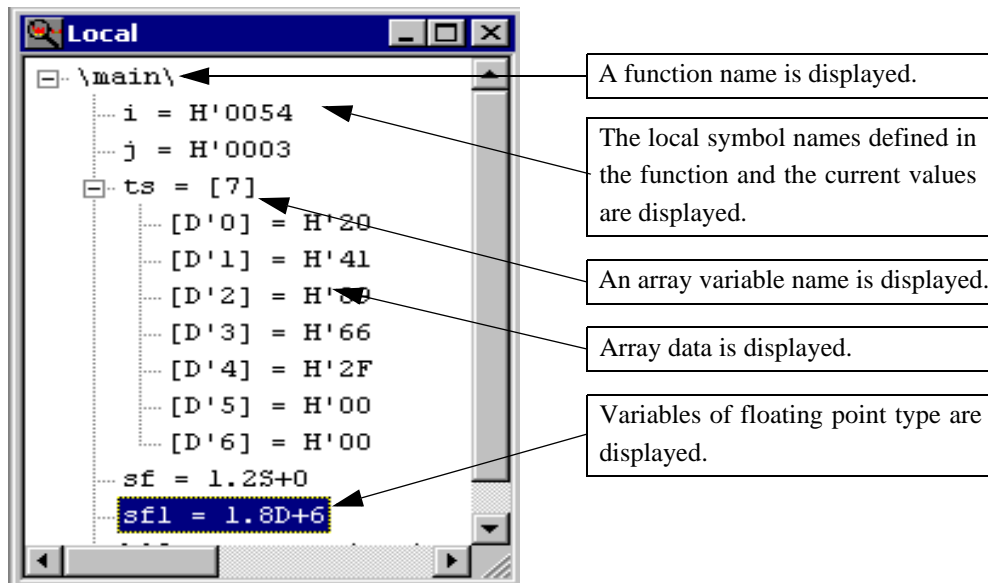
3.12 Local Variable Window

This window is displayed only when SOFTUNE Workbench is in the debug session. The local variable window is used to display and modify local variable values. Local variables are displayed in the tree format (function name root).

■ Local Variable Window

Figure 3.12-1 shows an example of the local variable window.

Figure 3.12-1 Example of Local Variable Window



■ Short-Cut Menu

- Radix: See Section "4.4.6 Local".
- Element: See Section "4.4.6 Local".
- Edit: See Section "4.4.6 Local".
- Memory window: Display the memory window and setup bookmark.
- Property: Displays symbol properties.
- Close: Closes local window.
- Memory window
 - When selected, the following two submenus are opened:
 - Jump
 - Display memory window by using selected address of variable.
 - Add bookmark
 - By using selected variable address, bookmark will be added to memory window.
 - For details, See Section "4.3.7 Bookmark".
- Property
 - Information for the variable is displayed. For details, see Section "4.3.9 Property".

3.13 Watch Window

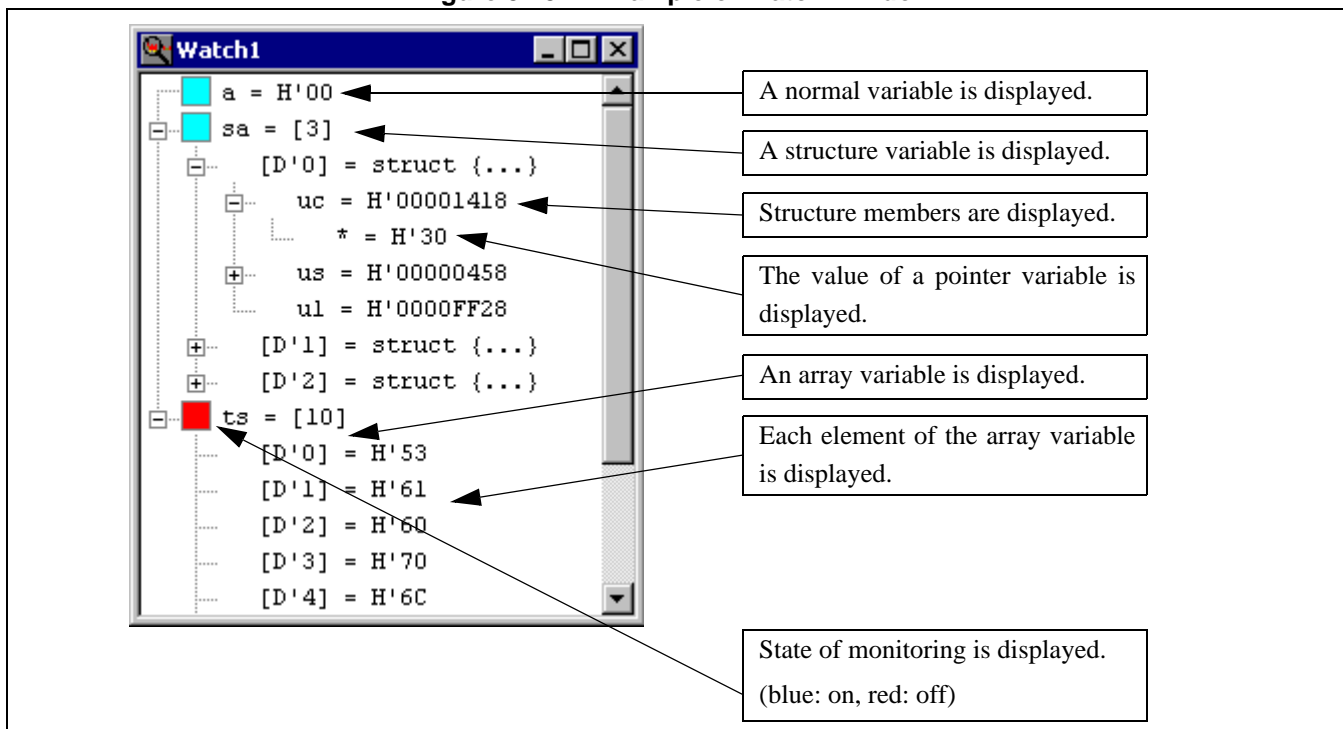
This window is displayed only when SOFTUNE Workbench is in the debug session. The watch window is used to display and modify the values of the specified variables. Variables are displayed in the tree format.

■ Watch Window

The watch window can be opened independent 4-windows. The watch window can be specified when the variables for each window is registered.

Figure 3.13-1 shows an example of the watch window.

Figure 3.13-1 Example of Watch Window



■ Function

- Drag and Drop
 1. Sort variables in the window
If you register multiple variables in the window, you can change the order of the variables by dragging a variable and dropping it into a desired place.
 2. Cooperation between the watch windows
You can copy/move the variables by dragging a registered variable and dropping it into another watch window. If you drag a structure member, only the member is copied to another window, and the structure remains in the original window.
- Multiple selection of variables
You can select multiple variables at a time for the batch operation. Available operations are as follows:
 - Copy/Move by drag and drop
 - Setting of the radix
 - Delete
 - On/Off of monitoring

■ Short-Cut Menu

- Radix: See Section "4.4.7 Watch".
If multiple variables are selected, they are deleted all together.
- Set: See Section "4.4.7 Watch".
- Element: See Section "4.4.7 Watch".
- Edit: See Section "4.4.7 Watch".
- Delete: See Section "4.4.7 Watch".
If multiple variables are selected, they are deleted all together.
- All Delete: Deletes all variables being stored from the window.
- Inaccessible area: Call up the [Inaccessible area] tab in the [Debug environment] of the [Setup] menu.
- Memory window: Display the memory window based on the selected variables.
- Monitoring: Switches whether to enable or disable monitoring.
- Individual setting: Each variable which is registered, user sets "ON" or "OFF" of monitoring. User distinguishes between "ON" and "OFF" by a color of square ICON. A blue means on, a red means off.
- Property: Displays watch properties.
- Close: Closes watch window.
- Property
 1. Click the right button of the mouse on a displayed variable name to display the short-cut menu, then select [Property] from the menu.
 - Information for the variable is displayed. For details, see Section "4.3.9 Property".
 2. Click the [Close] button

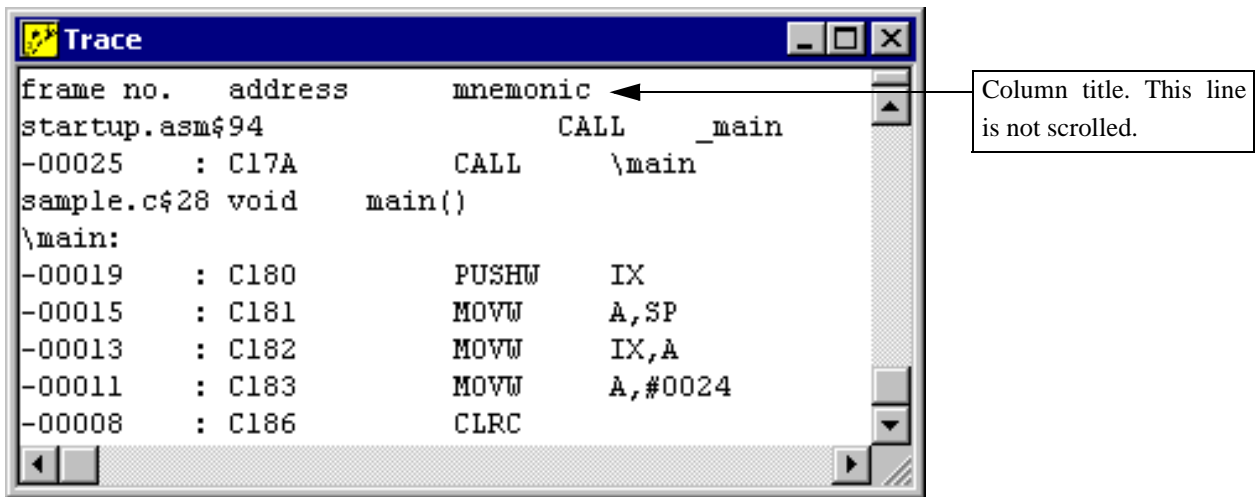
3.14 Trace Window

This section explains the Trace Window. This window is displayed only when SOFTUNE Workbench is in the debug session.

■ Trace Window

The Trace Window displays the trace result. For the trace function, see Section "4.4.8 Trace". Figure 3.14-1 shows an example of the Trace Window.

Figure 3.14-1 Example of Trace Window



■ Short-Cut Menu

- Refresh Refresh Trace Window.
 - Jump See Section "4.3.6 Jump".
 - Back Trace See Section "4.4.8 Trace".
 - Instruction Displays trace result in instruction mode.
 - Cycle Displays trace result in cycle mode (Emulator (MB2141)).
 - Source Displays trace result in source mode.
 - Setup See Section "4.4.8 Trace".
 - Find See Section "4.4.8 Trace".
 - Save See Section "4.4.8 Trace".
 - Clear Clears display mode trace buffer.
 - Close Closes Trace Window.
- Instruction/cycle/source
1. Click the right button of the mouse to display the short-cut menu.
 2. Select [Instruction], [Cycle], or [Source] from the short-cut menu.
 - The trace result is displayed in the selected mode and a check mark is displayed to the left of the selected submenu.

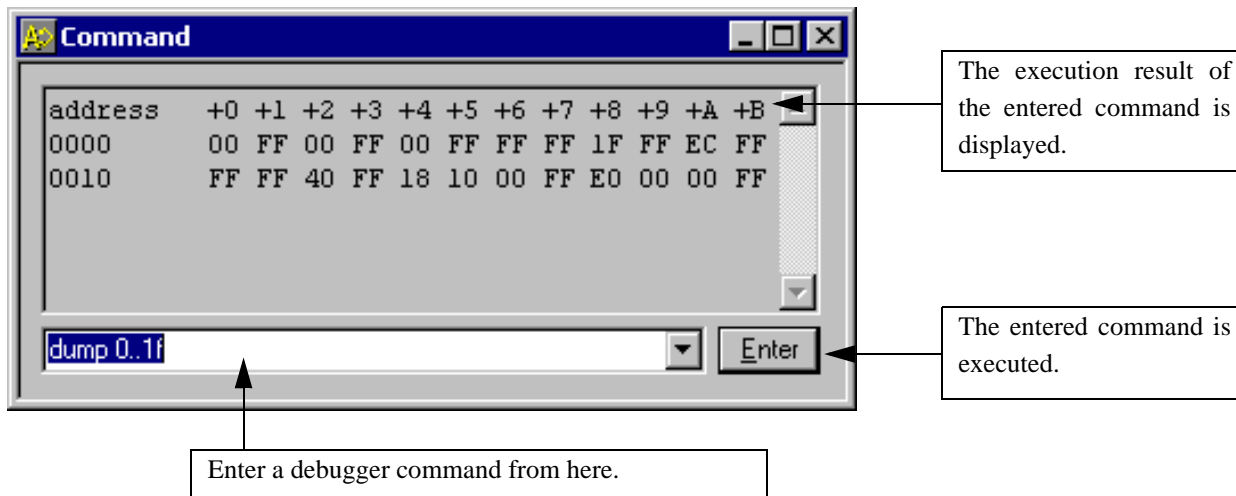
3.15 Command Window

This window is displayed only when SOFTUNE Workbench is in the debug session. The Command Window displays the execution result of the directly entered debugger command.

■ Command Window

Figure 3.15-1 shows an example of the Command Window.

Figure 3.15-1 Example of Command Window



■ Short-Cut Menu (Command Input Field)

- Undo Cancels the immediately preceding editing.
 - Cut Moves the selected character string to the clipboard.
 - Copy Copies the selected character string to the clipboard.
 - Paste Pastes the character string in the clipboard.
 - Delete Deletes the selected character string.
 - All select Selects all the entered character strings.
-
- Undo
 1. Click the right button of the mouse on the command input field to display the short-cut menu.
 2. Select [Undo] from the short-cut menu.
 - Cut
 1. Select the character string you want to cut.
 2. Click the right button of the mouse on the command input field to display the short-cut menu.
 3. Select [Cut] from the short-cut menu.
 - Copy
 1. Select the character string you want to copy.

2. Click the right button of the mouse on the command input field to display the short-cut menu.
3. Select [Copy] from the short-cut menu.
- Paste
1. Click the right button of the mouse on the command input field to display the short-cut menu.
2. Select [Paste] from the short-cut menu.
- Delete
1. Select the character string you want to delete.
2. Click the right button of the mouse on the command input field to display the short-cut menu.
3. Select [Delete] from the short-cut menu.
- All select
1. Click the right button of the mouse on the command input field to display the short-cut menu.
2. Select [All select] from the short-cut menu.

■ Short-Cut Menu (Other than the Command Input Field)

- | | |
|-------------------|--|
| - Load Batch File | Opens the file dialog for specifying the batch file. |
| - Load Alias File | Opens the file dialog for specifying the alias file. |
| - Macro etc | Displays Command Replacement Dialog. |
| - Logging | Sets log start, log status display and log end. |
| - Close | Closes Command Window. |
-
- Load Batch File
 1. Click the right button of the mouse to display the short-cut menu.
 2. Select [Load Batch File] from the short-cut menu.
The [Open File] file dialog opens.
 3. Select a batch file from the dialog, then click the [Open] button.
 - Load Alias File
 1. Click the right button of the mouse to display the short-cut menu.
 2. Select [Load Alias File] from the short-cut menu.
The [Open File] file dialog opens.
 3. Select an alias file from the dialog, then click the [Open] button.
 - Macro etc
 1. Click the right button of the mouse to display the short-cut menu.
 2. Select [Macro etc] from the short-cut menu.
The [Replace command] dialog opens. For details, see Section "3.15.1 Setting Character String Replacement".
 - Logging

For logging, see Section "3.15.2 Logging".
 - Close
 1. Click the right button of the mouse to display the short-cut menu.
 2. Select [Close] from the short-cut menu.

3.15.1 Setting Character String Replacement

This section explains how to set character string replacement in the Command Window.

■ Setting Character String Replacement

The aliases, variables, and macro names of the commands used in the Command Window can be defined.

■ Alias

The commands to be used often can be started simply by specifying their names. Set and delete aliases in the following procedures:

- Adding an alias

1. Click the right button of the mouse in the Command Window to display the short-cut menu.
2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

3. Set an alias.

Specify the alias that does not conflict with command names and the aliases that has been already registered.

4. Write the command to be assigned to the specified alias.
- The command can be written, including its parameters.

5. Click the [Append] button.

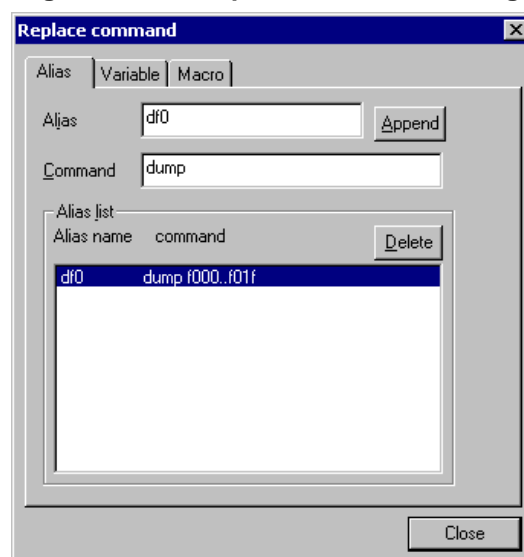
- Deleting an alias

1. Click the right button of the mouse in the Command Window to display the short-cut menu.
2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

3. Select the alias to be deleted from the alias list.
4. Click the [Delete] button.

Figure 3.15-2 Replace command Dialog



■ Variable

The name assigned to the address range to be referenced often can be used for command input. Add and delete a variable in the following procedures.

- Adding a variable

1. Click the right button of the mouse in the Command Window to display the short-cut menu.

2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

3. Click the [Variable] tag.

The variable setting dialog shown in Figure 3.15-3 opens.

4. Set a variable.

Specify the variable name that does not conflict with the variable names that have been already registered.

5. Write the character string to be assigned to the specified variable name.

6. Click the [Append] button.

- Deleting a variable

1. Click the right button of the mouse in the Command Window to display the short-cut menu.

2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

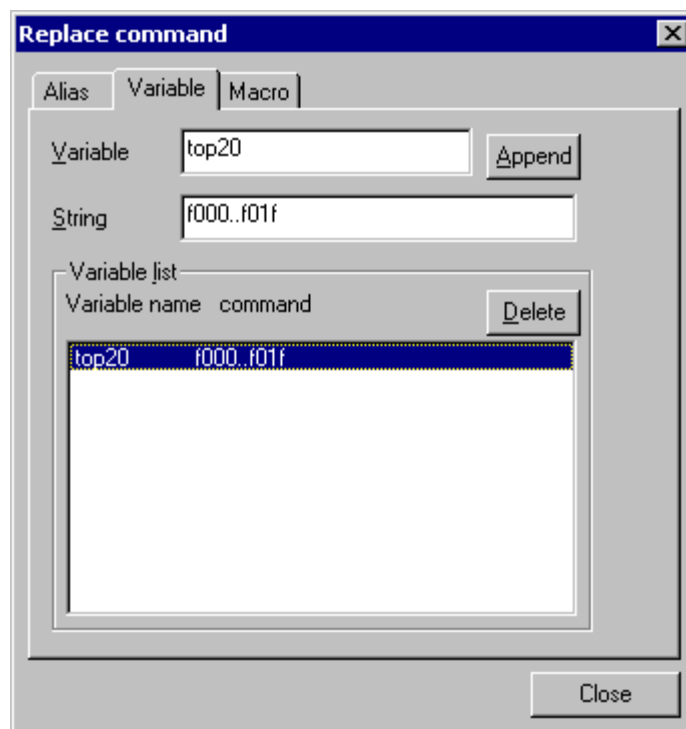
3. Click the [Variable] tag.

The variable setting dialog shown in Figure 3.15-3 opens.

4. Select the variable name to be deleted from the debug variable list.

5. Click the [Delete] button.

Figure 3.15-3 Variable Setting Dialog



■ Macro

A combination of the special commands to be used often can be named and used as a macro. Add and delete a macro in the following procedures:

- Adding a macro

1. Click the right button of the mouse in the Command Window to display the short-cut menu.
2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

3. Click the [Macro] tag.

The macro setting dialog shown in Figure 3.15-4 opens.

4. Set a macro name.

Specify a macro name that does not conflict with the macro names that have been already registered.

5. Write the commands to be assigned to the specified macro name.

Aliases, variables, and other macros can be also written.

6. Click the [Append] button.

- Deleting a macro

1. Click the right button of the mouse in the Command Window to display the short-cut menu.
2. Select [Macro etc] from the short-cut menu.

The character string replacement setting dialog shown in Figure 3.15-2 opens.

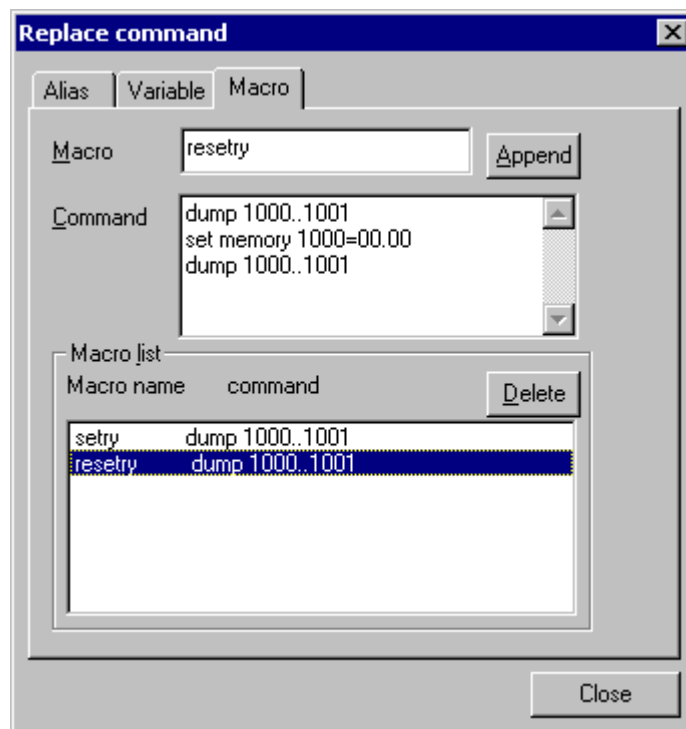
3. Click the [Macro] tag.

The macro setting dialog shown in Figure 3.15-4 opens.

4. Select the macro name to be deleted from the macro list.

5. Click the [Delete] button.

Figure 3.15-4 Macro Setting Dialog



3.15.2 Logging

This section explains SOFTUNE Workbench logging.

■ Logging

To record a command execution log in the Command Window in a file is called logging. To control logging, click the right button of the mouse in the Command Window to display the short-cut menu, then select [Logging].

Selecting [Logging] from the short-cut menu displays the following submenus:

- Start
- State
- Exit

- Start

1. Click the right button of the mouse to display the short-cut menu.
2. Select [Start] from the short-cut menu.
The file dialog for selecting a logging file opens.
3. Specify a logging file name, then click the [Save] button.

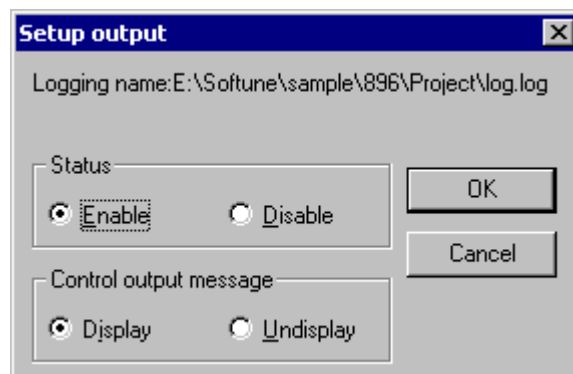
- State

1. Click the right button of the mouse to display the short-cut menu.
2. Select [State] from the short-cut menu.
The logging state display dialog shown in Figure 3.15-5 opens.
3. If necessary, change the state and output message control.
4. Click the [OK] button.

- Exit

1. Click the right button of the mouse to display the short-cut menu.
2. Select [Exit] from the short-cut menu.

Figure 3.15-5 Logging State Display Dialog



3.16 Object Window

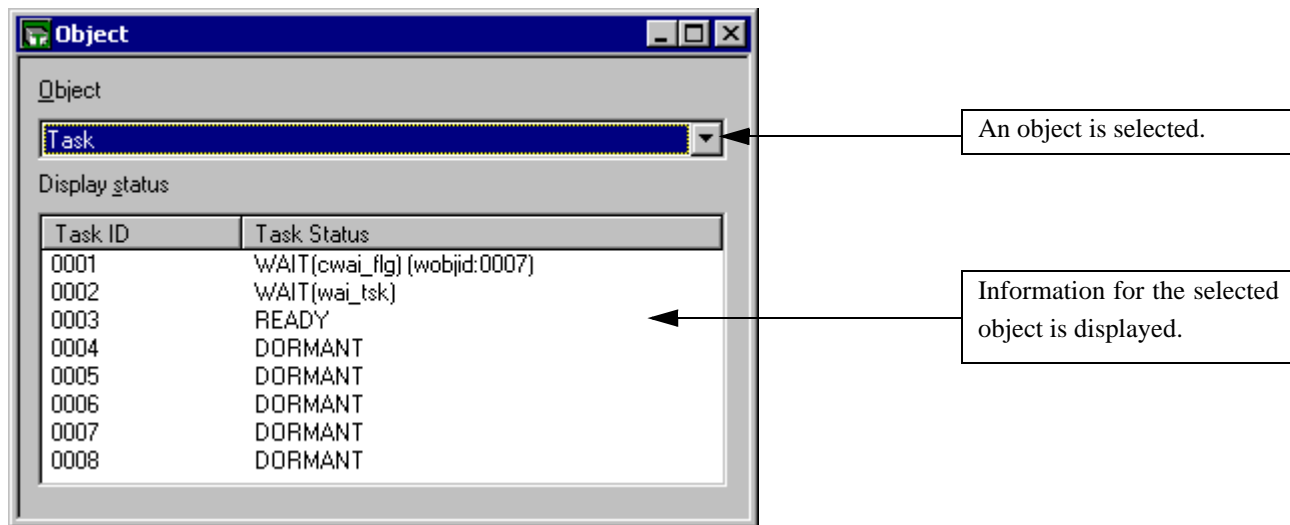
This window is displayed only when SOFTUNE Workbench is in the debug session. The Object Window displays the object information used by REALOS.

■ Object Window

The Object Window displays the object information used by REALOS.

Figure 3.16-1 shows an example of the Object Window.

Figure 3.16-1 Example of Object Window



■ Object/State Display

Object	Status display
Task	Task ID, task state
Semaphore	Semaphore ID, count of semaphores
Flag (1-bit flag)	Flag ID, SET or CLEAR
Mail box	Mail box ID, count of message queues, count of tasks
Cyclic handler	Handler ID, active state (residual time till activation)
Ready queue	Priority, count of tasks
Timer queue	Task ID or cyclic handler ID

■ Short-Cut Menu

- Task	Executes task system calls
- Semaphore	Executes semaphore system calls
- Event Flag	Executes event flag system calls
- Mail Box	Executes mail box system calls
- Ready Queue	Executes ready queue system calls
- Cyclic Handler	Executes cyclic handler system calls.
- Monitoring	Switches whether to enable or disable monitoring
- Property	Displays object properties
- Close	Closes Object Window

For details on the above items, see Section "4.4.11 Object".

3.17 Coverage Window

This window is displayed only when SOFTUNE Workbench is in the debug session. The Coverage Window displays the coverage measurement result.

■ Coverage Window

Figure 3.17-1 shows an example of the Coverage Window. The Coverage Window displays the coverage measurement result at debugging. This window is displayed only in MB2141 (emulator).
For details on the coverage function, see Section "4.4.12 Coverage".

Figure 3.17-1 Coverage Window (16-Address Unit)

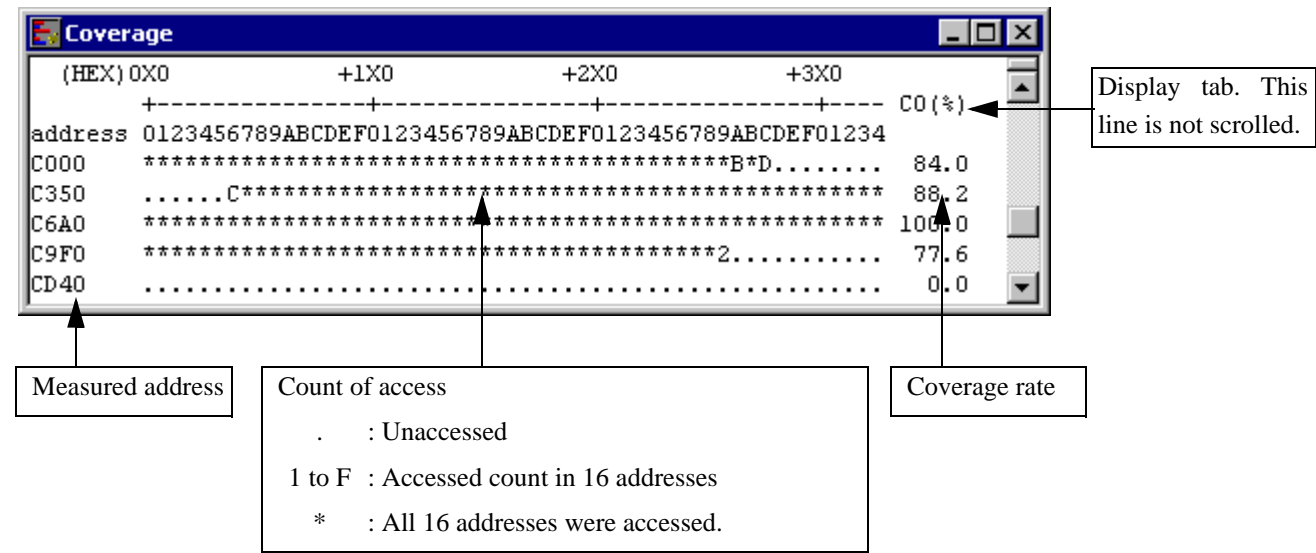
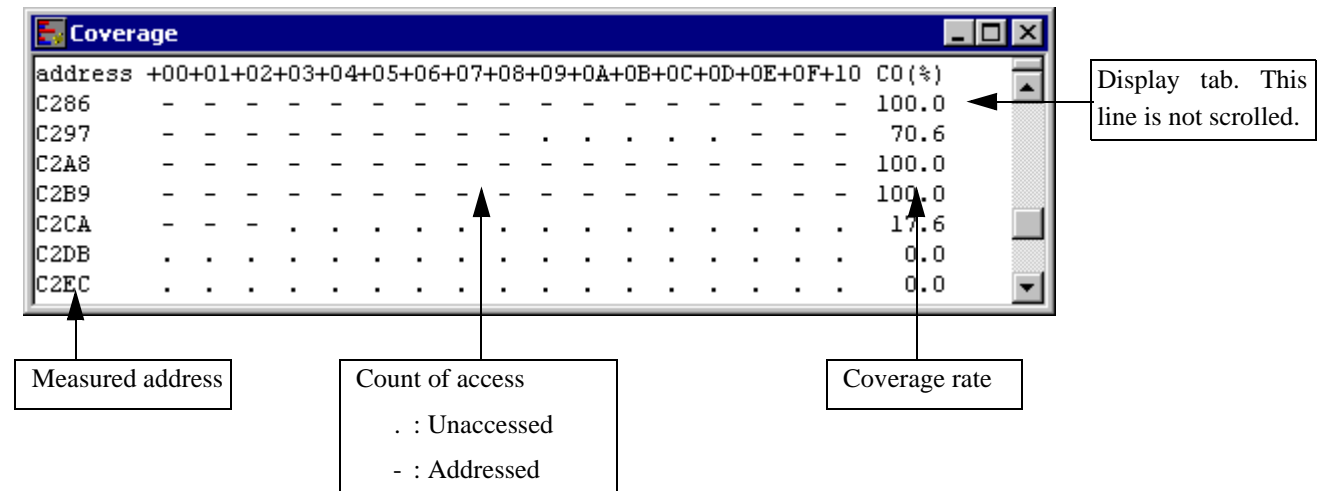


Figure 3.17-2 Coverage Window (1-Address Unit)



■ Function

- Drag and Drop

By dropping variable name, function name, label, or address from source window to window, display will be jumped to a location where an address of dropped character string.

■ Short-Cut Menu

- | | |
|-----------------------|--|
| - Refresh | Refresh Coverage Window display. |
| - Jump | Opens the jump dialog. |
| - Source window | Display the source window based on the address of cursor position. In case the line numbers corresponded to address are not existed, disassemble window will be displayed. |
| - 16 Address Unit | Displays the coverage measurement result in the unit of 16 addresses. |
| - 1 Address Unit | Displays the coverage measurement result for each address. |
| - Set | See Section "4.4.12 Coverage". |
| - Total Coverage Rate | See Section "4.4.12 Coverage". |
| - Clear | Clears the coverage measurement buffer. |
| - Close | Closes the Coverage Window. |

3.18 Performance Window

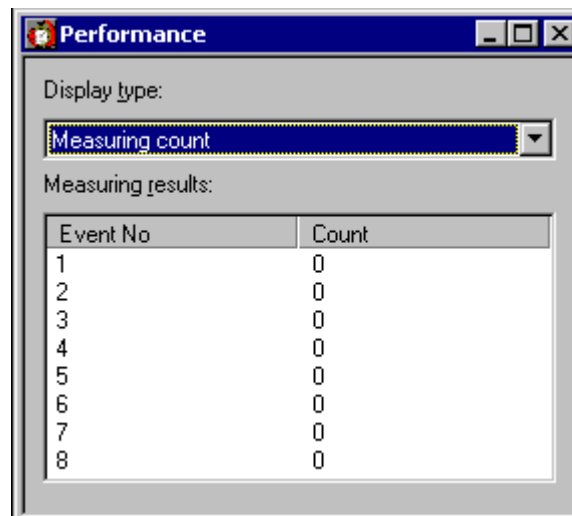
This window is only displayed for the emulator debugger (MB2141) and the performance mode at the event mode. The Performance Window displays the performance measurement result.

■ Performance Window

For the performance function, see Section "4.4.13 Performance".

Examples of the performance window are given in Figure 3.18-1 and Figure 3.18-2. The performance window is used to display the result of performance measurement at debugging. This window is only displayed for the emulator debugger (MB2141) and the performance mode at the event mode.

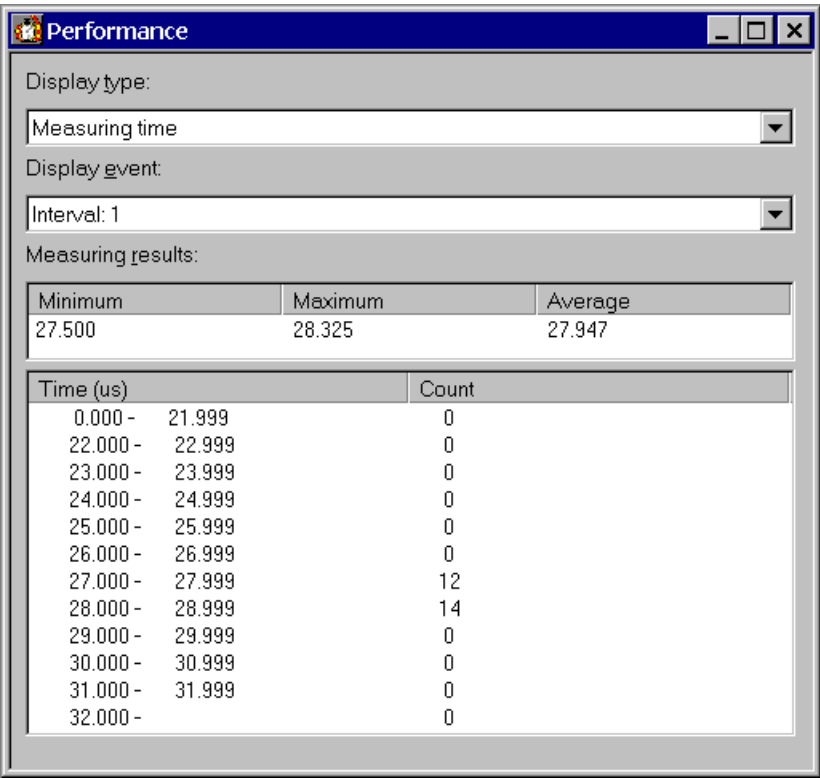
Figure 3.18-1 Performance Window (Measurement Count)



The screenshot shows a window titled "Performance" with a blue title bar. Inside, there is a "Display type:" label followed by a dropdown menu showing "Measuring count". Below this is a "Measuring results:" label followed by a table with two columns: "Event No" and "Count". The table contains 8 rows, each with an event number from 1 to 8 and a count of 0.

Event No	Count
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0

Figure 3.18-2 Performance Window (Measurement Time)



- Display Type Select measurement time or measurement count as the display type.
- Display Event A display event is displayed only when measurement time is selected.
Select a measurement interval.
- Measurement Result The measurement result is totaled and displayed.
The minimum measurement result, maximum measurement result, and average measurement result are displayed only when measurement time is selected as the display type.

■ Short-Cut Menu

- Refresh Refresh Performance Window display.
- Set See Section "4.4.13 Performance".
- Display Range See Section "4.4.13 Performance".
- Event See Section "4.6.5 Event".
- Clear Clears the Performance buffer.
- Copy See Section "4.4.13 Performance".
- Close Closes the Performance Window.

3.19 Sequence Window

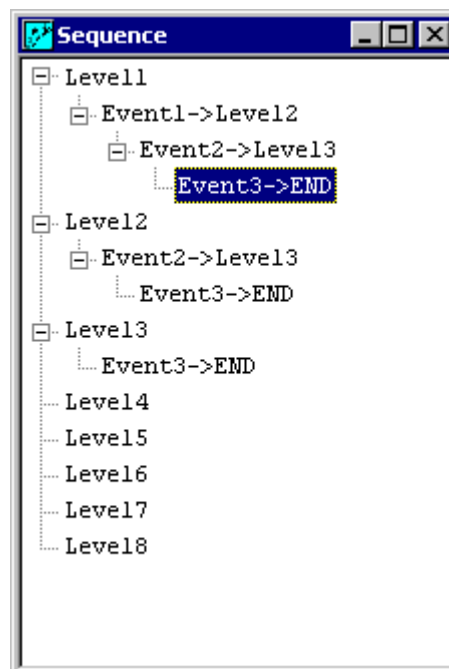
This window is displayed only when SOFTUNE Workbench is in the debug session. The Sequence Window displays the sequence setting state.

■ Sequence Window

For details on the sequence function, see Section "4.6.6 Sequence".

An example of the sequence window is shown in Figure 3.19-1. The sequence window is used to display the sequence setting state at debugging. This window is only displayed for the emulator (MB2141).

Figure 3.19-1 Sequence Window



■ Short-Cut Menu

- | | |
|-----------------|-------------------------------|
| - Setup | See Section "4.6.6 Sequence". |
| - Delete | Deletes sequence setting. |
| - Delay Count | See Section "4.6.6 Sequence". |
| - Set Latch | See Section "4.6.6 Sequence". |
| - Display Latch | See Section "4.6.6 Sequence". |
| - Event | See Section "4.6.5 Event". |
| - Close | Closes the Sequence Window. |

CHAPTER 4

MENUS

This chapter explains the SOFTUNE Workbench menu configuration and the dialogs to be started from each menu.

- 4.1 Menu Configuration (Tree)
- 4.2 File Menu
- 4.3 Edit Menu
- 4.4 View Menu
- 4.5 Project
- 4.6 Debug
- 4.7 Setup
- 4.8 Window
- 4.9 Help

4.1 Menu Configuration (Tree)

This section explains the SOFTUNE Workbench menu configuration.

■ Menu Configuration

The following menus are displayed in the SOFTUNE Workbench Main Window.

[File]	[Edit]	[View]	[Project]	[Debug]	[Setup]	[Window]	[Help]
--------	--------	--------	-----------	---------	---------	----------	--------

The following functions are assigned to each menu item.

- File
 - New
 - Open
 - Close
 - Open Workspace
 - Close Workspace
 - Save
 - Save As
 - Save All
 - Print
 - Recent Text File
 - Recent Workspace File
 - Exit
- Edit
 - Undo
 - Redo
 - Cut
 - Copy
 - Paste
 - Delete
 - All Select
 - Find
 - Replace
 - Find in files
 - Jump
 - Bookmark
 - Previous Error
 - Next Error
 - Top of Error

- Bottom of Error
- Property
- View
 - Project
 - Output
 - Symbol
 - Assembly
 - Register
 - Memory
 - Local
 - Watch
 - Trace
 - Coverage
 - Performance
 - Command
 - Object
 - Tool Bar
 - Status Bar
 - Bookmark
 - Fonts
- Project
 - Active Project
 - Add Project
 - Add Member
 - Setup Workspace
 - Setup Project
 - Setting Customized Build
 - Project Dependence
 - Project Configuration
 - Include Dependencies
 - Compile
 - Make
 - Build
 - Stop
- Debug
 - Run
 - Abort
 - Reset MCU
 - Break point

CHAPTER 4 MENUS

- Break Point Set/Reset
- Event
- Sequence
- Stack
- Time Measurement
- Call
- Clear Call
- Vector
- Load Target File
- Start Debug/Terminate Debug
- Setup
 - Development
 - Debug Environment
 - Memory Map
 - Flash area control
 - Tool
 - Keyboard
 - Editor
 - Error
 - Start Tool
- Window
 - Cascade
 - Vertical
 - Horizon
 - Split
 - Arrange Icons
 - Refresh
 - Refresh All Windows
 - Close All Windows
- Help
 - Help Topics
 - Support Information
 - About Fs896s

4.2 File Menu

File Menu of SOFTUNE Workbench has the following functions:

- **Workspace and file accessing**
 - **Printing**
 - **End of SOFTUE Workbench**
-

■ **Functions Related to Project and File Accessing**

The project and file accessing functions are as follows.

- New
- Open
- Close
- Open Workspace
- Close Workspace
- Save
- Save As
- Save All
- Recent Text File
- Recent Workspace File

■ **Other Functions**

Other functions are as follows.

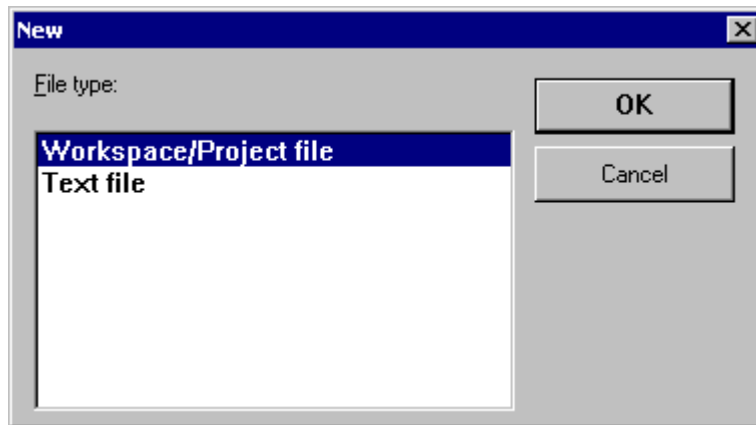
- Print
- Exit

4.2.1 New

"New" creates a workspace/project and file newly. Selecting "New" from the file menu opens the new creation dialog asking the operator whether to create a workspace/project or text file (Figure 4.2-1).

■ New Creation Dialog

Figure 4.2-1 New Creation Dialog

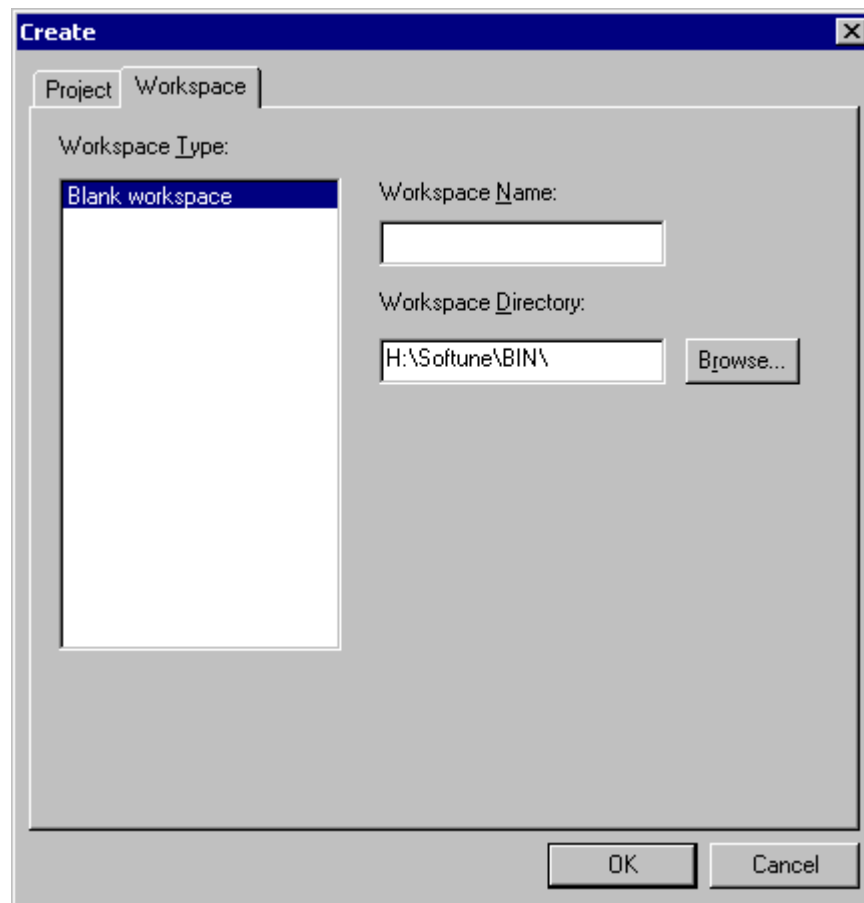


■ Creation of New Workspace

To create a new workspace, select [Workspace/Project File]. When the [New] dialog for project is opened, select the [Workspace] tab (Figure 4.2-2). Set the basic items regarding workspace to be created.

For details about the setting procedure, see Section "2.3 Creating Workspace".

Figure 4.2-2 Project New Creation Dialog (Workspace)

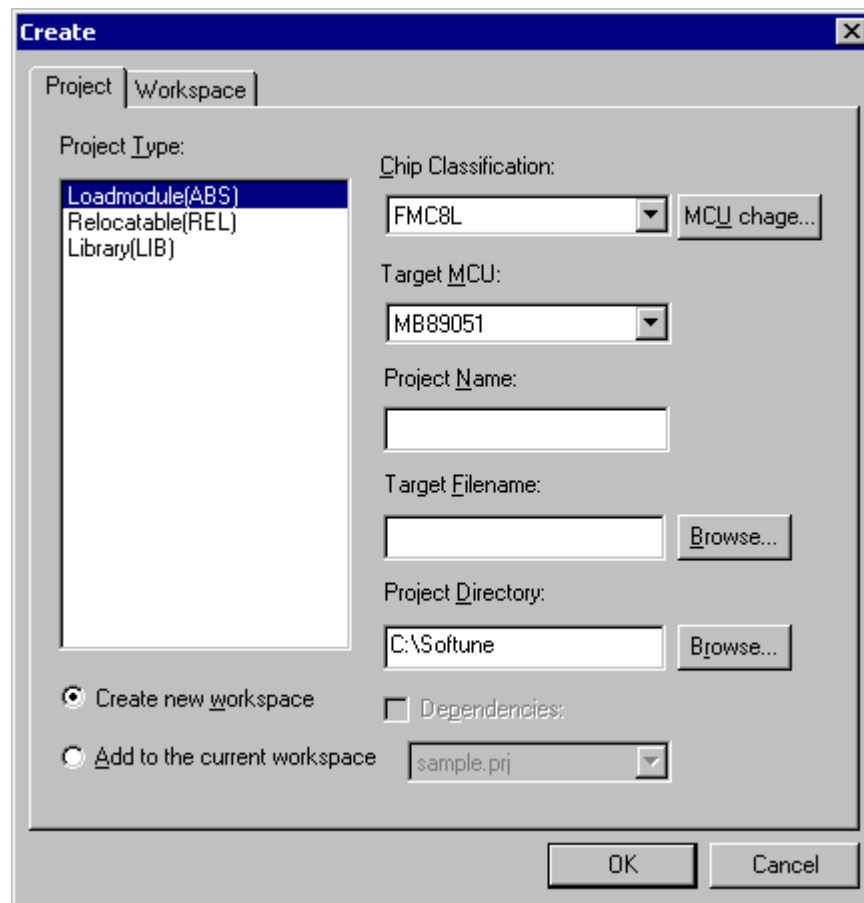


■ Creation of New Project

To create a new project, select [Workspace/Project File]. When the [New] dialog for project is opened, select the [Project] tab (Figure 4.2-3). Set the basic items regarding project to be created.

When creating a new workspace in creating a new project, see Section "2.3 Creating Workspace" and when adding to a existing workspace, see Section "2.4 Storing of Project".

Figure 4.2-3 Project New Creation Dialog (Project)



■ Creating Source File and Document Files, Etc.

To create a source file and document file, etc., select [Text File] from the new creation dialog. The editor is started in the status in which a new file is created.

4.2.2 Open

"Open" opens existing files. Selecting "Open" from the file menu opens the open dialog asking the file you want to open (Figure 4.2-4).

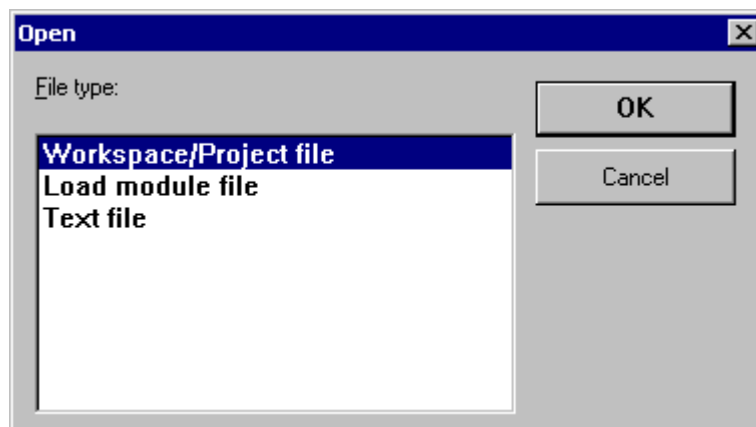
■ Open Dialog

Selecting [Open] from the file menu opens the open dialog.

The open dialog displays a workspace file, project file, load module file, source file, binary file, alias file, coverage file, batch file, and text file.

If SOFTUNE Workbench is not in the debug session, the binary file, alias file, coverage file, and batch file are not displayed.

Figure 4.2-4 Open Dialog



■ Opening the Load Module File

- When the project file has been already opened and SOFTUNE Workbench is in the debug session.
The file dialog for selecting the name of the load module to be debugged opens.
To load debug information only, set a check box for [Only Debug Information].
To set a memory map automatically during simulator debugger, set a check box for [Auto mapping]. If this check box is not set, set the memory map before opening the load module file.
- When the project file has been already opened but SOFTUNE Workbench has not been in the debug session.
The dialog asking the operator to make SOFTUNE Workbench enter the debug session opens. When the [OK] button is clicked, SOFTUNE Workbench automatically enters the debug session. Clicking the [Cancel] button cancels file open operation.
- When the workspace file is not opened.
The dialog for checking whether to create a new projects or whether to open the existing workspace/project opens.
 - To create a new projects, click the [Yes] button.
 - To read the existing workspace/project, click the [No] button.
 Clicking the [Cancel] button cancels file open operation.

■ **Opening the Binary File**

Specify the start address for binary data read in [Start Address].

■ **Opening the Batch File**

Setting [Run in Icon State] starts batch processing in the icon state.

4.2.3 Close

"Close" closes open files and windows.

■ Close

[Close] closes the files open in the Edit Window or debug windows (e.g., Register Window).

-Edit Window

When an unsaved modified file is in the Edit Window, the dialog for asking the operator whether to save the file opens.

-[Yes] button ... Saves the file using the existing file name.

-[No] button ... Does not save the file and closes the Edit Window.

-[Cancel] button ... Cancels the close operation.

-Debug windows (e.g., Register Window)

SOFTUNE Workbench closes debug windows immediately.

4.2.4 Open Workspace

"Open Workspace" opens the existing workspace file.

■ Opening the Existing Workspace File

The file dialog for selecting the file you want to open opens.

- When the existing workspace file has been already opened and SOFTUNE Workbench is in the debug session.

The dialog for asking the operator whether to terminate debugging opens.

-[OK] button... Terminates the debugger and continues workspace open processing.

-[Cancel] button... Cancels workspace open processing.

- When the existing workspace file has been already opened but SOFTUNE Workbench is not in the debug session.

The file dialog for selecting a workspace file opens. When a workspace file name is specified, SOFTUNE Workbench closes the currently open workspace file and then opens the specified file. If the open registered file is not saved although it is being edited, the dialog for asking the operator whether to save the file opens.

-[Yes] button... Saves the file using the existing file name and continues processing.

-[No] button... Continues processing without saving the file.

-[Cancel] button... Cancels workspace open processing.

- When the existing workspace file is not open.

The file dialog for selecting a workspace file opens. When a workspace file name is specified, SOFTUNE Workbench opens the specified file. When a workspace file is opened, the used window when the file was saved is redisplayed.

-[Cancel] button... Processing is continued with the file being edited opened.

The project file can be opened instead of the workspace file. For details, see Section "2.3 Creating Workspace".

4.2.5 Close Workspace

"Close Workspace" closes the currently open workspace file.

■ When the Currently Open Workspace File is not Edited at All

When the workspace file is closed, the dialog for asking the operator whether to save the current workspace information opens.

- | | |
|------------------|---|
| -[Yes] button | ... Saves the current workspace information and closes the workspace file. |
| -[No] button | ... Closes the workspace file without saving the current workspace information. |
| -[Cancel] button | ... Cancels workspace file close processing. |

In the following cases, however, the above dialog does not open:

- The workspace file and project are not modified.
- When the workspace file is closed, inquiry for save is not set.

■ When the Currently Open Workspace File is being Edited

The dialog for asking the operator whether to save the current workspace information opens. When the [Yes] or [No] button is clicked to continue processing, the dialog for asking the operator whether to save the file being edited subsequently opens.

- | | |
|------------------|--|
| -[Yes] button | ... Saves the file being edited and then closes the workspace file. |
| -[No] button | ... Closes the file being edited without saving the file being edited. |
| -[Cancel] button | ... Workspace is closed with the file being edited opened. |

4.2.6 Save

"Save" saves the currently open file using the existing file name.

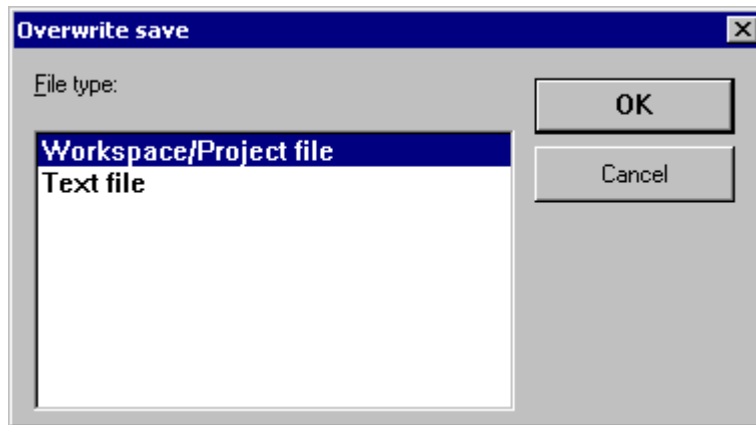
■ Save Dialog

The save dialog is used to save the file using the existing file name.

The text file, workspace file, binary file, alias file, setup file, or coverage file can be selected from this dialog as the type of file.

If SOFTUNE Workbench is not in the debug session, the binary file, alias file, setup file, and coverage file are not displayed.

Figure 4.2-5 Save Dialog



4.2.7 Save As

"Save As" stores the previously saved file under a new file name.

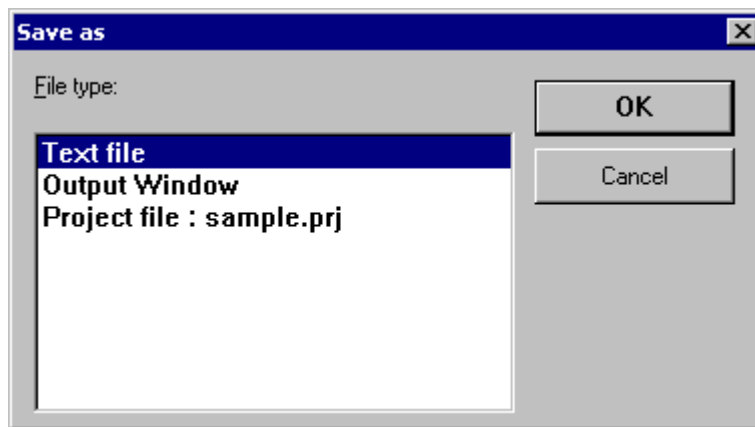
■ Save As

[Save As] stores the previously saved file under a new file name.

The text file, output window, project file, binary file, alias file, setup file, or coverage file can be selected as the type of file.

If SOFTUNE Workbench is not in the debug session, the binary file, alias file, setup file, and coverage file are not displayed.

Figure 4.2-6 Save as Dialog



■ Saving the Project File

Save the active project file.

Select the format for saving the project file from the [File Type] combo box.

For the project formats, refer to Section "1.2 Management Function for Project" of SOFTUNE Workbench User's Manual.

If a Workspace project format (*.prj) is selected, it saves all project information in workspace project format (compatible to workspace).

When the project file opened in old project format is overwritten in workspace project format, conversion is performed and restrictions on old project files are cleared. Subsequent saving is performed in workspace project format.

If an Old project format (*.prj) is selected, it saves information on active project configurations in old project format (incompatible to workspace).

The project file opened in workspace project format cannot be overwritten in old project format. In this case, save the file as a different name.

■ Saving the Binary File

Specify a save start address in [Start Address]. When the binary file has been already loaded, the start address at that time is set.

Specify a save end address in [End Address]. When the binary file has been already loaded, the end address at that time is set.

Note:

If the project file in a different directory from the original one is saved, the members described in a relative path may not be referred. In this case, move the members with the original directory configuration kept being unchanged.

4.2.8 Save All

"Save All" saves all the files currently being edited.

■ Files to be Saved

The files to be saved include text files such as the source file opened in the edit window. Data in the workspace, project file and output window is not saved.

Even the text file opened in the edit window is not saved if it is not edited.

When SOFTUNE Workbench is in the debug session, the binary, alias, setup, or coverage file that was edited even once is to be saved.

■ Processing a Newly Created Text

If there is a newly created text that has not been saved yet, the file dialog for asking the operator to enter the name of the file to be saved opens. Specify a file name from this dialog.

4.2.9 Print

"Print" prints file data.

■ Data to be Printed

[Print] prints data in the file currently opened in the edit window; it cannot print other window (file) data.

Print output window data in any of the following two procedures:

- Copy output window data to the edit window and print it.

Copy output window data to the edit window in the following procedure, then select [Print] from the [File] menu:

1. Select all output window data.
2. Click the right button of the mouse to display the menu, then select [Copy].
3. Open the edit window, then select [Paste] from the [Edit] menu.

- Save output window data once, then print it.

For how to save output window data, see Section "4.2.7 Save As".

Open the saved file with the edit window, then select [Print] from the [File] menu.

4.2.10 Recent Text File/Recent Workspace File

Of the text files opened in the Source Window in the debugger, "Recent Text File" stores up to five most recently opened text files. Of the used workspace files, "Recent Workspace File" stores up to five most recently opened workspace files.

■ Recent Text File

Of the text files opened in the Source Window, [Recent Text File] stores up to five most recently opened text files. It enables to open the desired text file immediately by selecting it.

The files in the Edit Window are not stored here.

■ Recent Workspace File

Of the used workspace files, [Recent Workspace File] stores up to five most recently opened workspace files. It enables you to open the desired workspace file immediately by selecting it.

Note:

If a text or project file has been already deleted or moved to another directory, it cannot be opened even if it is listed here.

4.2.11 Exit

"Exit" terminates SOFTUNE Workbench.

■ When a File is being Edited

The dialog for asking the operator whether to save the file being edited opens.

- | | |
|-----------------|--|
| [Yes] button | ... Saves the file being edited and terminates SOFTUNE Workbench. |
| [No] button | ... Terminates SOFTUNE Workbench without saving the file being edited. |
| [Cancel] button | ... Does not terminate SOFTUNE Workbench. |

■ When the Workspace is Opened

The dialog for asking the operator whether to save the current workspace information opens.

- | | |
|-----------------|--|
| [Yes] button | ... Saves the current workspace information in the file. |
| [No] button | ... Does not save the current workspace information in the file. |
| [Cancel] button | ... Does not terminate SOFTUNE Workbench. |

If no file is being edited, SOFTUNE Workbench terminates immediately. If there is a file being edited, processing explained in "When a file is being edited" is performed.

■ When SOFTUNE Workbench is in the Debug Session

As in processing explained in "When the workspace is opened", the dialog for asking the operator whether to save the current workspace information opens. Clicking [Yes] or [No] from this dialog continues processing and automatically terminates the debug session.

If no file is being edited, SOFTUNE Workbench terminates immediately. If there is a file being edited, processing explained in "When a file is being edited" is performed.

4.3 Edit Menu

The edit menu provides such functions as file editing, find string, and error jump.

■ File Editing Function

The file editing function includes the following:

- Undo
- Redo
- Cut
- Copy
- Paste
- Delete
- All Select

■ Character String Search Function

The character string search function includes the following:

- Find
- Replace
- Find in Files

■ Jump Function

Enables to move to any place in a window quickly:

- Jump

■ Bookmark Function

Enables to setup a bookmark or to jump to the bookmark:

- Bookmark

■ Error Jump Function

The error jump function includes the following:

- Previous Error
- Next Error
- Top of Error
- Bottom of Error

■ Property Function

Display the information of activated window:

- Property

4.3.1 Undo, Redo

"Undo" cancels the immediately preceding editing and undoes the status before the editing.

Further "Redo" can undo the deleted editing to the previous status.

■ Target

The function is available only for edits for the edit window. It can also continuously cancel or undo a "series of actions" for successive addition or deletion of characters and lines.

Note:

This function is valid only for the edit window. Note that the value changed in the Memory or Register Window when SOFTUNE Workbench is in the debug session cannot be undone. [Undo], [Redo] function may not always work for a large amount of edits.

4.3.2 Cut, Copy, Paste, Delete

"Cut" cuts the selected character string and "Copy" copies it to the clipboard. "Delete" deletes the selected character string. "Paste" inserts the cut or copied character string into the cursor position or replaces it with the selected character string.

■ Cut

"CUT" cuts the character string selected from the edit window and moves it to the clipboard. This command can be executed only in the edit window.

The character string cut here can be pasted later.

■ Copy

"COPY" copies the character string selected from the edit window or the Symbol Window, Assembly Window, Register Window, Memory Window, Local Symbol Window, Watch Window, Trace Window, Command Window or Coverage Window (valid in debug session) to the clipboard.

The character string copied here can be pasted later.

■ Paste

"PASTE" inserts the cut character string or the character string copied to the clipboard into the current cursor position in the edit window. If the selected character string is in the edit window, this command also replaces it with the character string in the clipboard.

■ Delete

"DELETE" deletes the character string selected from the edit window. This command can be executed only in the edit window.

The deleted character string cannot be pasted later.

Table 4.3-1 Relationship between Edit Functions and Windows

Window	Cut	Copy	Paste	Delete
Edit Window	○	○	○	○
Project Window	○	×	×	×
Output Window	×	○	×	×
Symbol Window	×	○	×	×
Assembly Window	×	○	×	×
Register Window	×	○	×	×
Memory Window	×	△ *1	×	×
Local Symbol Window	×	△ *2	×	×
Watch Window	×	△ *2	×	×
Trace Window	×	△ *1	×	×
Command Window	×	○	×	×
Coverage Window	×	△ *1	×	×
Performance Window	×	×	×	×
Sequence Window	×	×	×	×
Object Window	×	×	×	×

*1: Lines other than the column title can be copied.

*2: Only the symbol name can be copied.

4.3.3 All Select

"All Select" selects all the character strings displayed in a window.

■ **Target**

This function operates only in the edit and output windows.

4.3.4 Find/Replace

"Find" searches the text file displayed in the edit window for any character string.
"Replace" replaces the found character string with the specified character string.

■ Target

"FIND" and "REPLACE" can be executed only for the text files displayed in the edit window; they cannot be executed in the output window and others.

■ Find

When the find dialog shown in Figure 4.3-1 opens, specify the character string you want to find from this dialog. Character string search conditions (Table 4.3-2) can also be specified from this dialog.

The found character string is displayed in reverse video.

The find dialog is not automatically closed irrespective of whether the character string is found. For this reason, when search terminates, click the [Cancel] button to close the find dialog.

When SOFTUNE Workbench is in the debug session, the Source Window can also be searched.

Figure 4.3-1 Find Dialog



■ Replace

When the replace dialog shown in Figure 4.3-2 opens, specify a search character string and a replacing character string. In this case, character string search conditions (Table 4.3-2) can be also specified from this dialog.

The found character string is displayed in reverse video. Clicking the [Replace] button replaces the specified search character string with the specified replacing character string. Clicking the [All Replace] button replaces all the character strings found before search terminates.

The replace dialog is not automatically closed irrespective of whether the found character string was replaced. For this reason, when search terminates, click the [Cancel] button to close the replace dialog.

Figure 4.3-2 Replace Dialog

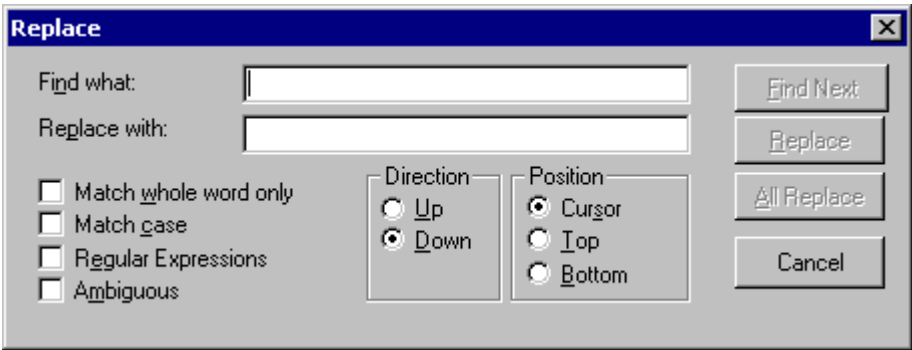


Table 4.3-2 List of Conditions and Explanation for Character String Search

Condition	Explanation
Match whole word only	Does not search partial character strings of words. Effective in searching for independent words that may become parts of other words such as "able".
Match case	Distinguishes uppercase characters from lowercase characters or vice versa.
Regular Expression	Specifies a search character string in regular expression. Enables use of a wild card, etc.
Direction	Specifies a search direction (Up or Down). "Up" searches the file upward. "Down" searches the file downward.
Position	Specifies a search start position (cursor, top, bottom).

4.3.5 Find in Files

This section explains how to search the specified file for the specified character string.

■ Find in Files

SOFTUNE Workbench searches the specified file for the specified character string and displays the search result in the output window.

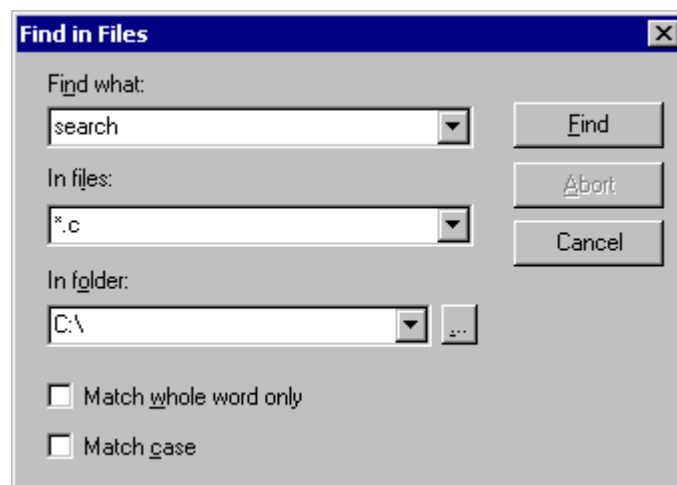
SOFTUNE Workbench can search several files at a time.

A line can be immediately edited by double-clicking the mouse on the line in the output window.

■ Search Procedure

1. Select the [Edit] - [Find in Files] menu.
Dialog shown in Figure 4.3-3 opens.
2. Specify a character string.
The dialog showing the character strings in the clipboard opens.
3. Specify the file to be searched.
A wild card can also be used to specify the file to be searched.
4. Specify the directory to be searched.
Click the button to the right of the specified field to open the directory search dialog.
5. Specify [Match whole word only] and [Match case] as required.
6. Click the [Find] button.
The search result appears in the output window in real time.
7. To abort a search, click [Abort] button.
The file search is aborted.
8. When search terminates, click the [Cancel] button.

Figure 4.3-3 Find in Files Dialog



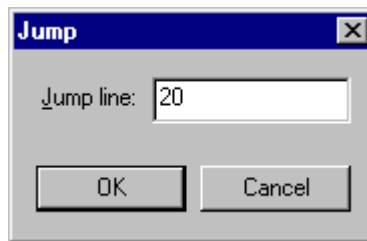
4.3.6 Jump

"Jump" moves the cursor to any line in the text file being edited. When SOFTUNE Workbench is in the debug session, this function (command) specifies the display start position of a source line, disassemble, memory, or trace.

■ Edit Window

When the jump dialog shown in Figure 4.3-4 opens, specify a jump destination line number. When the edit window is active, the cursor in the edit window jumps to the specified line irrespective of whether SOFTUNE Workbench is in the debug session.

Figure 4.3-4 Jump Dialog (Edit)



■ Debug Session

If a window other than the edit window becomes active when SOFTUNE Workbench is in the debug session, the dialog shown in Figure 4.3-5 opens. Select [TYPE], then specify [POSITION] in the selected format.

If [Address] is specified in [Type], [window] at source, memory or disassemble window can be used to specify the window where jump is implemented.

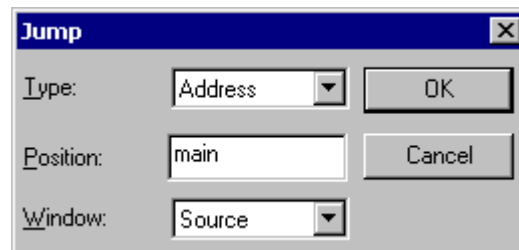
The following can be selected as [TYPE].

- Line number
- Address
- Frame

The following can be selected as [WINDOW].

- Source Window
- Memory Window
- Disassemble Window

Figure 4.3-5 Jump Dialog (Debug)



4.3.7 Bookmark

It is convenience in setting a bookmark for referring the location specified for source, memory and edit windows often. Once a bookmark is set, there will be a mark on specified location. This enables to jump to the specified location by using menu or key operation. Bookmark will be validated until it is canceled.

■ Bookmark

Bookmark is a function to specify the location in window and enable to jump to the location by using menu or key operation.

■ Compliant Window

Bookmark is valid in following windows.

- Source window
- Memory window
- Edit window

■ Bookmark Menu

Menu related to bookmark is located at [Edit]-[Bookmark] in a menu.

Submenu consists of 7 types as follows. Menu becomes valid when it is in debug session or each window is activated.

- Bookmark
- Setup/cancel bookmark
- Next bookmark
- Previous bookmark
- Next bookmark in the current window
- Previous bookmark in the current window
- Clear all bookmarks in the current window

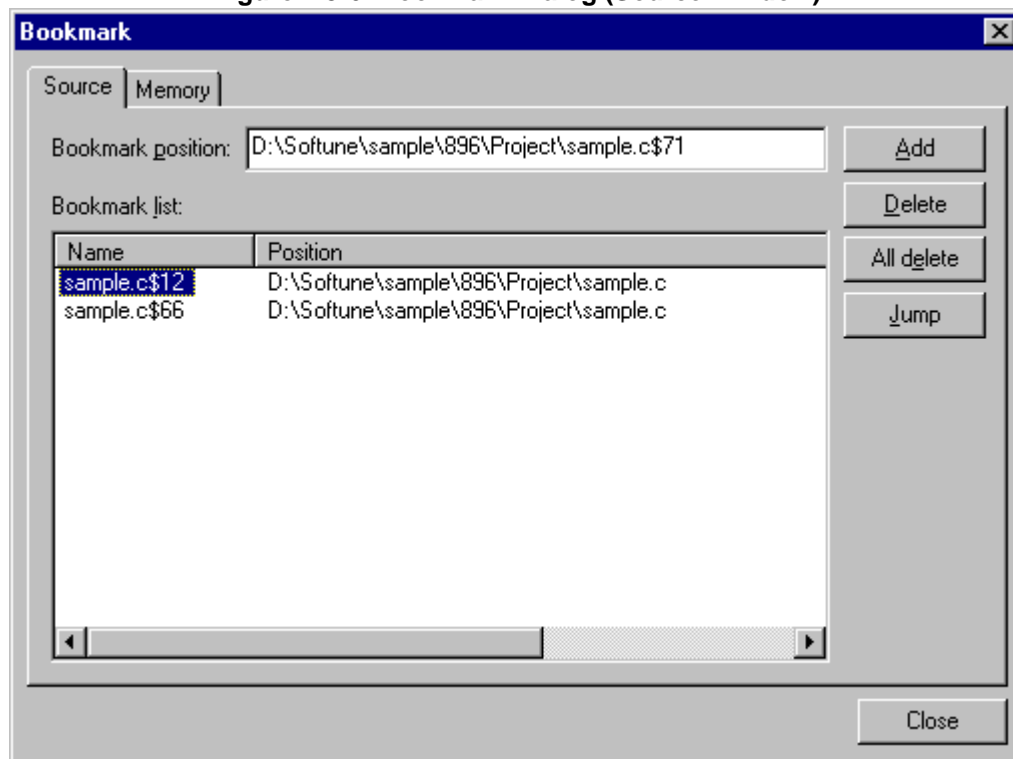
4.3.7.1 Bookmark - Source window

This section explains the operation when source window is activated.

■ To Display a Dialog "Bookmark"

By selecting [Edit] - [Bookmark] - [Bookmark], dialog as shown in Figure 4.2-6 will be opened. This dialog is to indicate the list of bookmark set in source window.

Figure 4.3-6 Bookmark Dialog (Source Window)



● Bookmark position

- This is a field to specify the position to set a bookmark. Entering relative path, function name or label name into the field can specify position of bookmark.

● Bookmark list

- The list of bookmark, which is currently set in a source window, will be displayed.
 - Name: Name of bookmark will be displayed. Name of bookmark will be appeared as line number or symbol name.
 - Position: File name, of which bookmark is set, will be displayed.
 - Line: Line number, where bookmark is set, will be displayed.
 - Symbol: Symbol, which is allocated to bookmark name, will be displayed. If a symbol is allocated, Bookmark position will be updated based on debug information although the line number of symbol changes along with the change in file. Please refer to [Note] for allocatable symbol.

● [Add] button

- Set a bookmark to a location specified by [Bookmark position]

● [Delete] button

- Delete selected bookmark from a list.

● [All delete] button

- Delete all bookmarks from a bookmark list.

● [Jump] button

- Move a start position for displaying a source window to location of bookmark selected in bookmark list.

■ **Setup or Cancel of a Bookmark**

- 1) Please put a cursor to line where you want to set or cancel a bookmark.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Setup/cancel bookmark]
 - Click on [Setup/cancel bookmark] at toolbar [Find]
 - Add or delete bookmark to/from [Source] tab in dialog [bookmark]

■ **Move the Caret to the Next Bookmark**

- 1) Please make sure that there is a cursor in a source window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Subsequent bookmark]
 - Click on [Subsequent bookmark] at toolbar [Find]
- 3) If no subsequent bookmark is exist on a window, proceed to the subsequent bookmark located on the next file.

■ **Move the Caret to the Previous Bookmark**

- 1) Please make sure that there is a cursor in a source window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Previous bookmark]
 - Click on [Previous bookmark] at toolbar [Find]
- 3) If no previous bookmark is existed on a window, proceed to previous bookmark located on the last file.

■ **Move the Caret to the Next Bookmark in the Current Window**

- 1) Please make sure that there is a cursor in a source window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Next bookmark in the current window]
 - Click on [Next bookmark in the current window] at toolbar [Find]

■ **Move the Caret to the Previous Bookmark in the Current Window**

- 1) Please make sure that there is a cursor in a source window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Previous bookmark in the current window]
 - Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear All Bookmarks in the Current Window

- 1) Please make sure that there is a cursor in a source window.
 - 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Clear all bookmarks in the current window]
 - Click on [Clear all bookmarks in the current window] at toolbar [Find]
 - Click on [All delete] at tab [Source] in a dialog [Bookmark]
-

Note:

Symbol, which can not be allocated to the name of bookmark, will be member of automatic variables, structure, union, and class.

If some symbols with the same name are defined, other symbol than assumed one may be used. To specify a symbol with the same name, please determine the name of module and function name first, then specify a symbol.

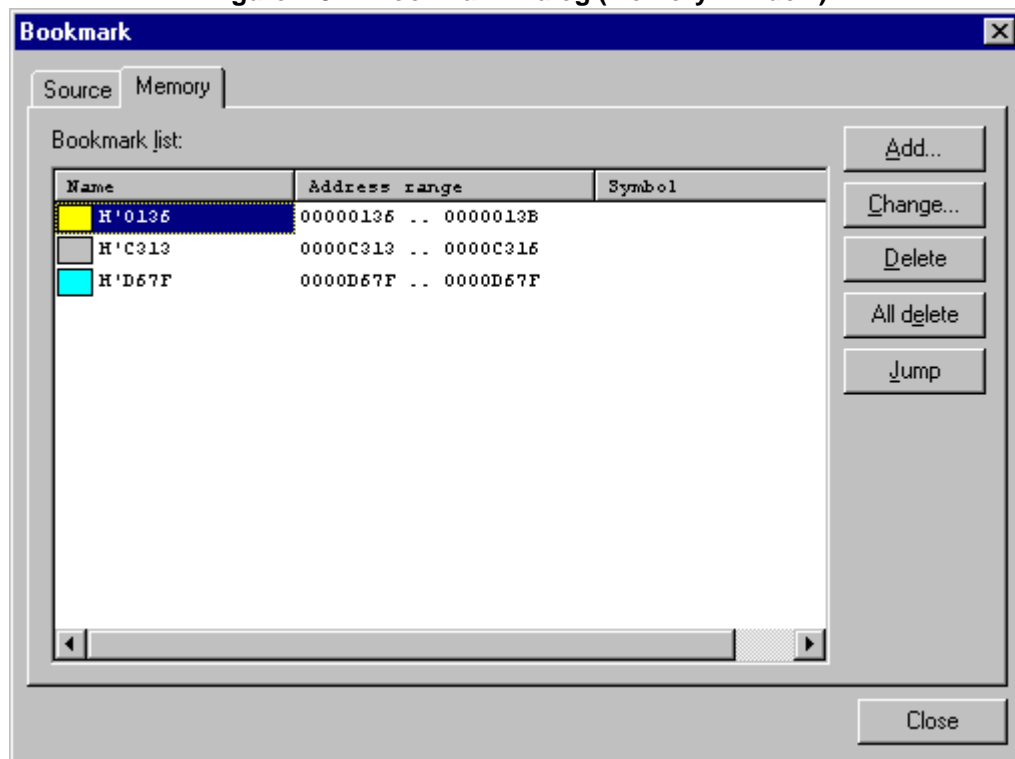
4.3.7.2 Bookmark - Memory Window

This section explains the operation when memory window is activated.

■ To Display a Dialog "Bookmark"

By selecting [Edit] - [Bookmark] - [Bookmark], dialog as shown in Figure 4.3-7 will be opened. This dialog is to indicate the list of bookmark set in memory window.

Figure 4.3-7 Bookmark Dialog (Memory Window)



● Bookmark list

- The list of a bookmark, which is currently set in memory window, will be displayed.

Name: Name of a bookmark and color of marking will be displayed. Address or symbol name will be name of bookmark.

Address range: Range of an address where bookmark is set is displayed.

Symbol: Symbol allocated to a bookmark will be displayed. If a symbol is allocated, location of bookmark will be updated based on debug information although the line number of symbol changes along with the change in file. Please refer to [Note] for allocatable symbol.

● [Add] button

- Display dialog [Add a bookmark] to add a bookmark. For details, refer to Section "4.4.5 Memory".

- [Change] button

- Display [Add a bookmark] dialog and change the setup of selected bookmark in a bookmark list.

- [Delete] button

- Delete a bookmark selected from the list of a bookmark.

- [All Delete] button

- Delete all bookmarks from the list of a bookmark.

- [Jump] button

- Move a start position for displaying a memory window to position of bookmark selected in bookmark list.

■ Setup or Cancel of a Bookmark

- 1) Please select a range of an address where you want to set or to cancel a bookmark
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Setup/cancel bookmark]
 - Click on [Setup/cancel bookmark] mark at toolbar [Find]
 - Setup a book mark with short-cut menu [Add a bookmark] (For details, refer to Section "4.4.5 Memory")
 - Add or delete a book mark at tab [Memory] in the Bookmark dialog

■ Move the Caret to the Next Bookmark in the Current Window

- 1) Please make sure that there is a cursor in a memory window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Next bookmark in the current window]
 - Click on [Next bookmark in the current window] at toolbar [Find]

■ Move the Caret to the Previous Bookmark in the Current Window

- 1) Please make sure that there is a cursor in a memory window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Previous bookmark in the current window]
 - Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear all Bookmarks in the Current Window

- 1) Please make sure that there is a cursor in a memory window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Clear all bookmarks in the current window]
 - Click on [Clear all bookmarks in the current window] at toolbar [Find]
 - Click on [All delete] at tab [Memory] in the Bookmark dialog

Note:

Symbol, which can not be allocated to the name of bookmark, will be member of automatic variables, structure, union, and class.

If some symbols with the same name are defined, other symbol than assumed one may be used. To specify a symbol with the same name, please determine the name of module and function name first, then specify a symbol.

4.3.7.3 Bookmark - Edit window

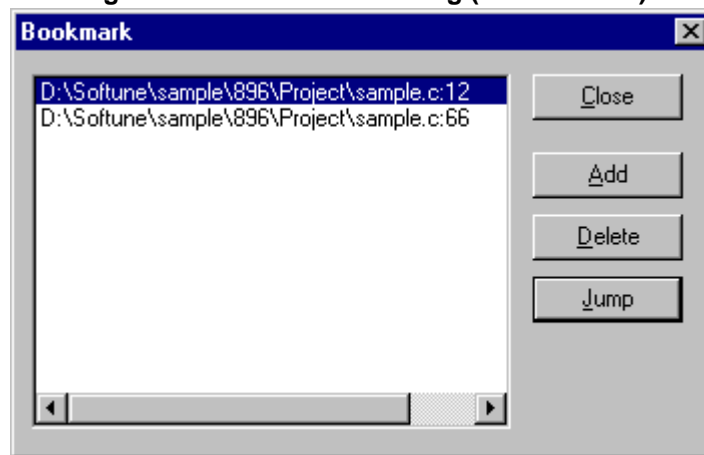
This section explains the operation when edit window is activated

■ To Display a Dialog "Bookmark"

By selecting one of following menus, dialog as shown in Figure 4.3-8 will be opened. This dialog is to display the list of bookmark that is set to an edit window.

1. [Edit] - [Bookmark] - [Bookmark]
2. Short-cut menu of Edit window [Bookmark]

Figure 4.3-8 Bookmark Dialog (Edit Window)



● Bookmark list

- The list of a bookmark (file name, line number), which is set currently to an edit window, will be displayed.

● [Add] button

- Setup of a bookmark to location of an edit window where cursor is put.

● [Delete] button

- Delete selected bookmark from a list. When more than one bookmarks are selected, delete all items.

● [Jump] button

- Move a start position for displaying a edit window to location of bookmark selected in bookmark list.

■ Setup or Cancel of a Bookmark

- 1) Please move a cursor to line where you want to set or cancel a bookmark.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Setup/cancel bookmark]
 - Click on [Setup/cancel bookmark] mark at toolbar [Find]
 - Click [Add] or [Delete] button at the Bookmark dialog.

■ Move the Caret to the Next Bookmark in the Current Window

- 1) Please make sure that there is a cursor in a edit window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Next bookmark in the current window]
 - Click on [Next bookmark in the current window] at toolbar [Find]

■ Move the Caret to the Previous Bookmark in the Current Window

- 1) Please make sure that there is a cursor in a edit window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Previous bookmark in the current window]
 - Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear all Bookmarks in the Current Window

- 1) Please make sure that there is a cursor in a edit window.
- 2) Please execute any of following operations.
 - Select [Edit] - [Bookmark] - [Clear all bookmarks in the current window]
 - Click on [Clear all bookmarks in the current window] at toolbar [Find]

4.3.8 Previous error, Next error, Top of Error, Bottom of Error

"Previous error", " Next error", "Top of Error" and "Bottom of Error" moves the cursor in the edit window to the source line where a make, build, compilation, or assembler error occurred.

■ Previous Error

The cursor jumps to the error line immediately before the cursor position in the output window. If there is no error line before the cursor position, the cursor loops back and jumps to the bottom error line.

■ Next Error

The cursor jumps to the error line immediately after the cursor position in the output window. If there is no error line after the cursor position, the cursor loops back and jumps to the top error line.

■ Top of Error

The cursor jumps to the top error line displayed in the output window.

■ Bottom of Error

The cursor jumps to the bottom error line displayed in the output window.

4.3.9 Property

"Property" displays information for the specified file.

■ Property

PROPERTY displays file information, module information, and symbol information. The information to be displayed depends on the currently active window.

There are six windows below.

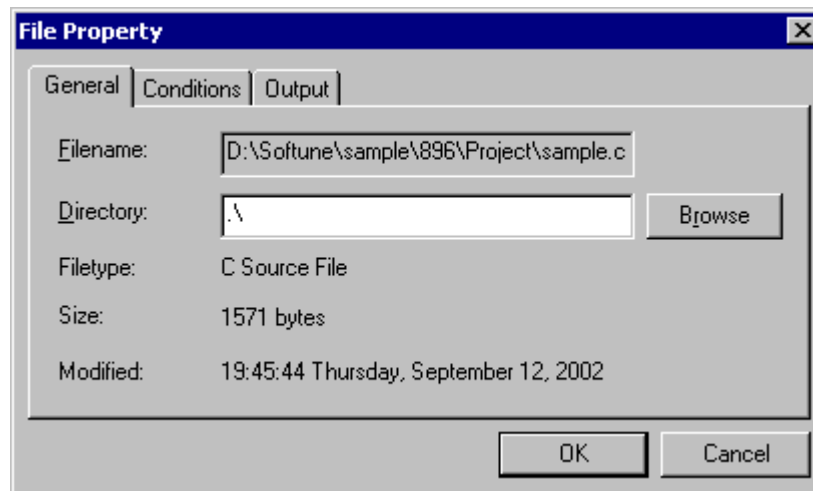
- Project Window
- Edit Window
- Source Window
- Symbol Window
- Local Window
- Watch Window

4.3.9.1 Property-Project Window

This section explains the property in the Project Window.

■ Property: General

Figure 4.3-9 Property-Project Window (General 1)



-File name

The full-path to the selected file is displayed.

-Directory

The directory of the selected file is displayed. If editing is enabled, the file to be referred can be changed.

If given in a relative path, the directory is described in a relative path in the project file.

-File type

The type of file managed in the project is displayed.

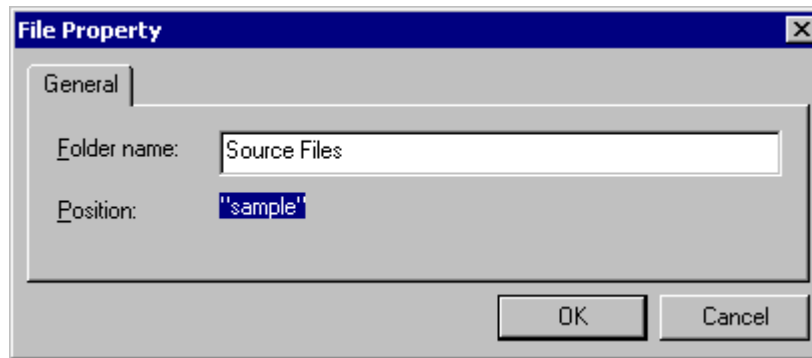
-Size

Displays the file size.

-Modified

Displays the date and time of the last file update.

Figure 4.3-10 Property-Project Window (General 2)



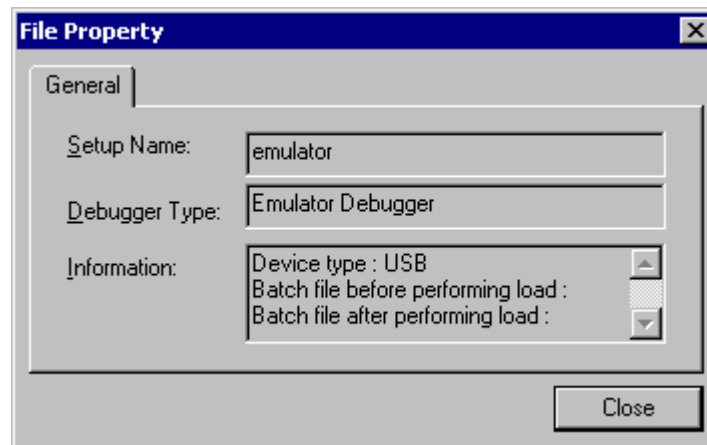
-Folder name

The name of the selected folder is displayed. If editing is enabled, folder name can be changed.

-Position

The place of the selected folder is displayed.

Figure 4.3-11 Property-Project Window (General 3)



-Setup name

The selected setup name is displayed.

-Debugger type

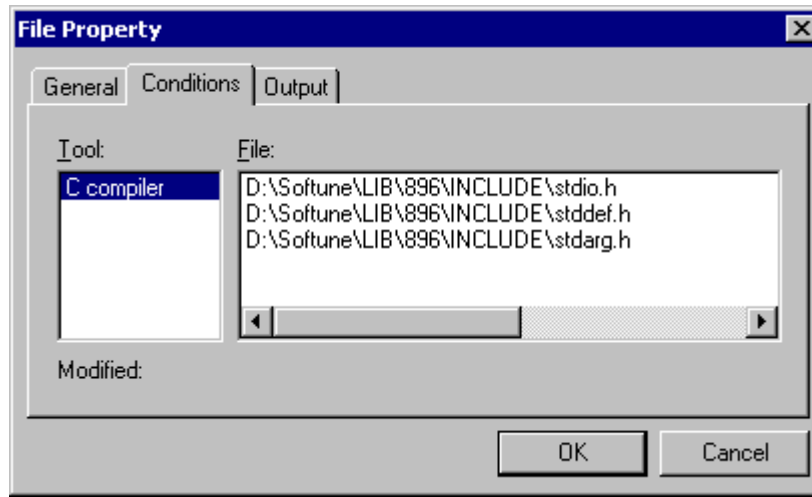
The Debugger type set in Debugger setup is displayed.

-Information

The device type, host, port, baud rate, batch file before performing load, batch file after performing load, etc., are displayed. The displayed information depends on the device type, etc.

■ Property: Conditions

Figure 4.3-12 Property-Project Window (Dependence)



-Tool

Displays the language tool to be executed.

-File

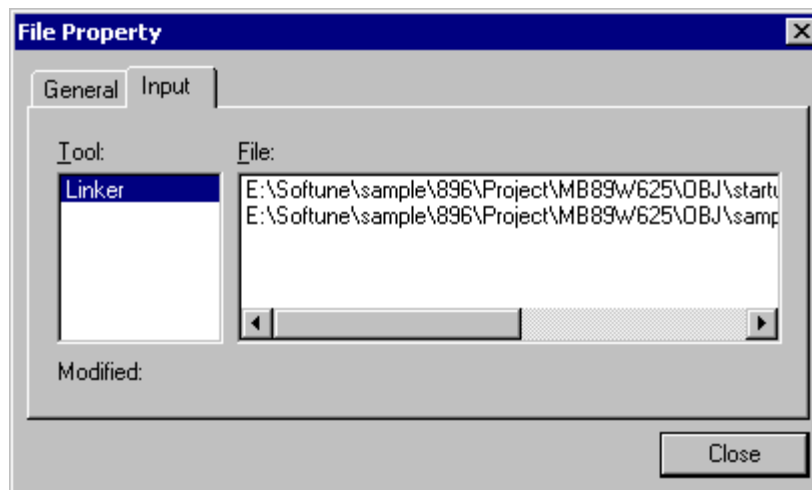
Displays a dependency file list. Selecting a file name from the list displays the update date and time of the file.

-Modified

Displays the date and time of the last file update.

■ Property: Input

Figure 4.3-13 Property-Project Window (Input)



-Tool

Displays the language tool to be executed.

-File

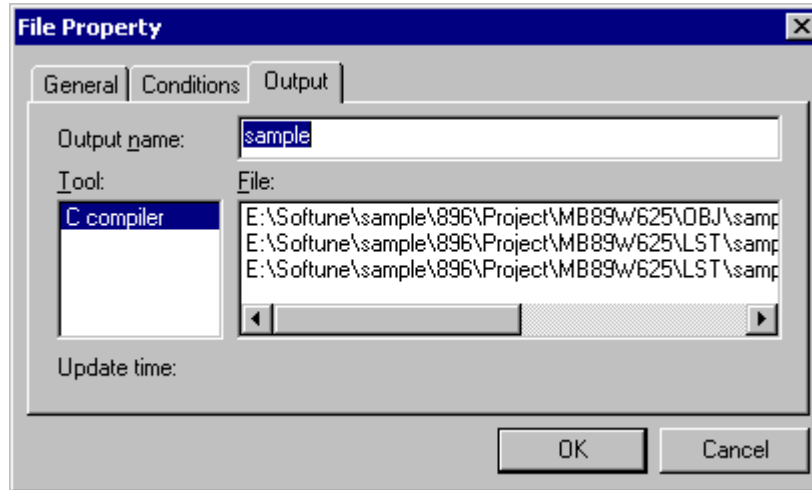
The file used in creating a target file is displayed in order.

-Modified

Displays the date and time of the last file update.

■ Property: Output

Figure 4.3-14 Property-Project Window (Output)



-Output name

Main file name of all files outputted by the language tool of the selected file is displayed.

-Tool

Displays the language tool to be executed.

-File

The file outputted by the language tool of the selected file is displayed.

-Updating time

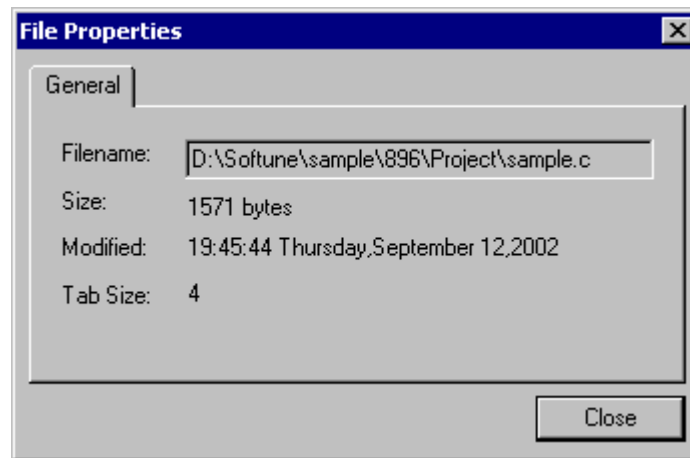
Displays the date and time of the last file update.

4.3.9.2 Property-Edit Window

This section explains the property in the Edit window.

■ Property: General

Figure 4.3-15 Property-Edit Window (General)



-File name

The full-path to the selected file is displayed.

-Size

Displays the file size.

-Modified

Displays the date and time of the last file update.

-Tab size

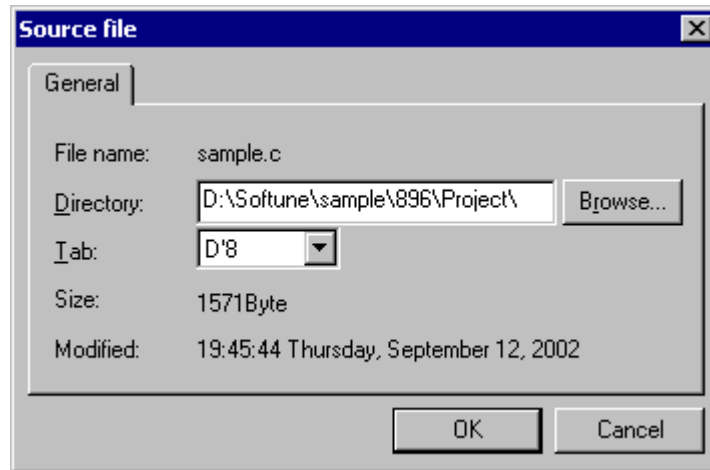
The size of tab code is displayed

4.3.9.3 Property-Source Window

This section explains the property in the Source window.

■ Property: General

Figure 4.3-16 Property-Source Window (General)



-File name

The full-path to the selected file is displayed.

-Directory

Displays the directory in which the source file exists. When changed, the directory is searched and the found source file is redisplayed.

-Tab

Displays the tab size. It is possible to change.

-Size

Displays the file size.

-Modified

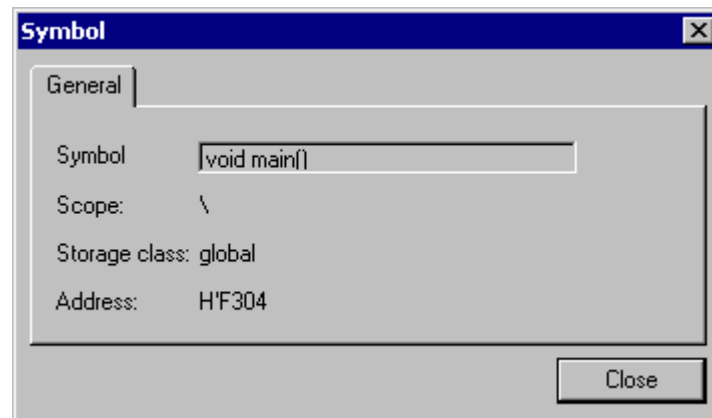
Displays the date and time of the last file update.

4.3.9.4 Property-Symbol Window

This section explains the property in the Symbol window.

■ Property: General

Figure 4.3-17 Property-Symbol Window (General)



-Symbol name

The name of the selected symbol is displayed.

-Scope

The scope position of the selected symbol is displayed.

-Storage class

The storage class of the selected symbol is displayed.

-Address

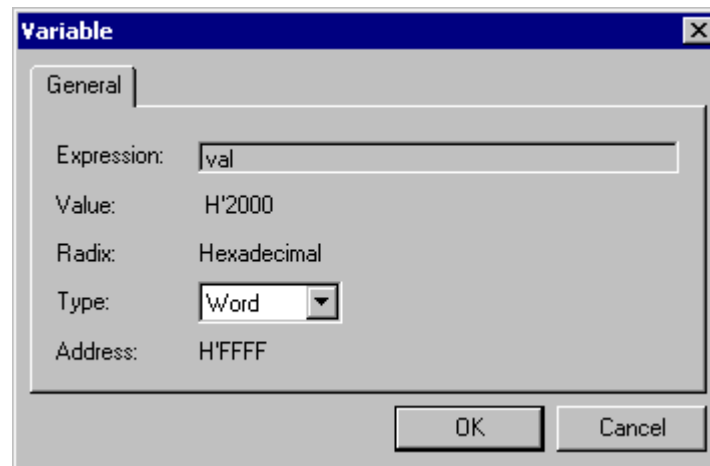
The address of the selected symbol is displayed.

4.3.9.5 Property-Local Window

This section explains the property in the Local window.

■ Property: General

Figure 4.3-18 Property-Local Window (General)



-Expression

The selected item is displayed.

-Value

The value of the selected expression is displayed.

-Radix

The base in which the value is given is displayed.

-Type

The type of the selected expression is displayed.

-Address

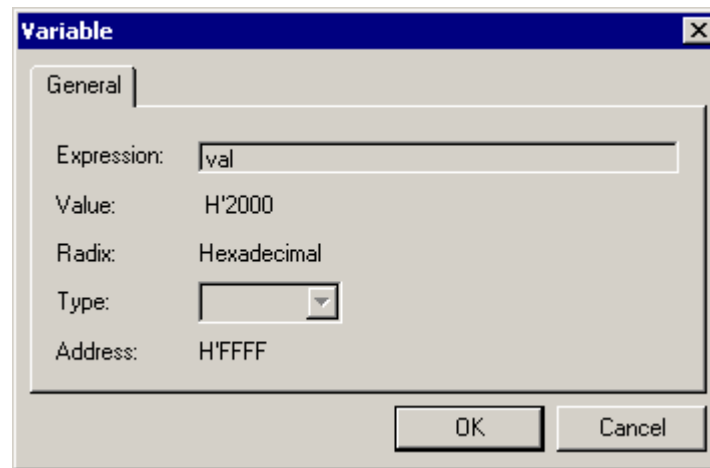
The symbol-address of the selected expression is displayed.

4.3.9.6 Property-Watch Window

This section explains the property in the Watch window.

■ Property: General

Figure 4.3-19 Property-Watch Window (General)



-Expression

The selected item is displayed.

-Value

The value of the selected expression is displayed.

-Radix

The base in which the value is given is displayed.

-Type

The type of the selected expression is displayed.

-Address

The symbol-address of the selected expression is displayed.

4.4 View Menu

The View Menu displays each window. It also sets the tool bar and status bar to view or non-view.

■ Window View/Non-view

The View Menu can set the following windows to view or non-view.

- Project Window
- Output Window

■ Window Display (Debug Session)

The View Menu displays the following Debugger windows (valid only when SOFTUNE Workbench is in the debug session):

- Symbol
- Assembly
- Register
- Memory
- Local
- Watch
- Trace
- Coverage
- Performance
- Command
- Object

■ Tool Bar/Status Bar

The View Menu can set the tool bar and status bar to view or non-view. The set of tool buttons displayed in the tool bar can be also selected using the View Menu.

- Tool Bar
- Status Bar

■ Font-related Items

The fonts for each window can be changed.

- Fonts

■ Display/non-display the Window Switching Using the Tab

This function enables to choose whether the tab is displayed or non-displayed for switching. By using the switching tabs, it enables to switch edit, source, memory or watch windows easily.

- Tab

4.4.1 Project/Output

"Project" switches the Project Window to view or non-view or vice versa. "Output" switches the Output Window to view or non-view or vice versa.

■ Project Window

When the Project Window is displayed, a check mark is set to the left of [Project] in the View Menu. Even if the Project window is switched to non-view, the project is not closed. When it is switched to display again, the Project Window is displayed at the previously displayed position.

■ Output Window

When the Output Window is viewed, a check mark is set to the left of [Output Window] in the View Menu. Even if the Output Window is switched to non-view, data in the Output Window is not cleared. When it is switched to displayed again, the Output Window is displayed at the previously displayed position.

If make, build, compile, assemble, update of dependencies or stop is executed when the Output Window is set to non-view, the Output Window automatically enters the view status.

Even if the Output Window is set to non-view when make, build, compile, or assemble is being executed, error messages are never lost. These messages are all displayed when the Output Window is redisplayed.

4.4.2 Symbol

"Symbol" views the Symbol Window.

■ Symbol

When the Symbol Window is opened, all the symbols used in the target file are displayed in the tree format.

This function (command) can only be used when SOFTUNE Workbench is in the debug session.

Selecting [Symbol] when the Symbol Window has been already opened activates the Window.

4.4.3 Assembly

"Assembly" displays the Disassemble Window.

■ Assembly

When the Assembly window is opened, the assembly is displayed, starting at the specified address. Line assembly from the short-cut menu is also possible.

This function (command) can be used only when SOFTUNE Workbench is in the debug session.

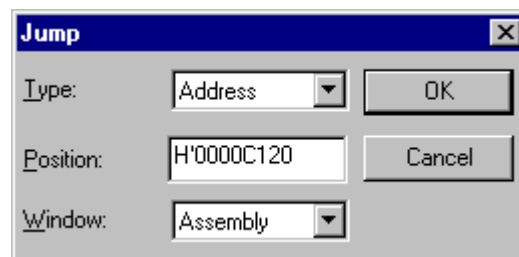
- When the Assembly Window has been already opened

The Assembly Window is activated.

- When the Assembly Window is not opened

The dialog for specifying the display start address (Figure 4.4-1) opens. Specify the address where display is to be started, then click the [OK] button.

Figure 4.4-1 Dialog for Specifying Display Start Address



- Type

Specify the type of display start position (Line number/Address/Frame).

- Position

Display disassembling from the position specified in the above type.

- Window

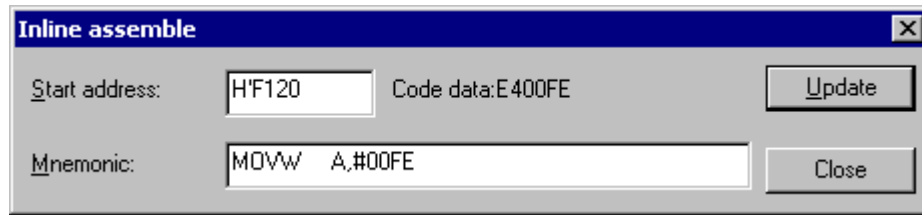
To specify the window where jump is implemented. For details, see Section "4.3.6 Jump".

If it is a disassemble window, default will be shown as [Assembly].

■ Inline Assemble

Selecting [Inline Assemble] from the short-cut menu opens the Inline assemble dialog shown in Figure 4.4-2.

Figure 4.4-2 Inline Assemble Dialog



Writing a mnemonic in the [Mnemonic] edit box and clicking the [Update] button assembles and sets the mnemonic, starting from the start address. The start address subsequently advances to the next address. To change the address where the mnemonic is to be written, change [Start Address]. When mnemonic change is completed, click the [Close] button.

4.4.4 Register

"Register" displays the Register Window.

■ Register

When the Register Window is opened, the selected target MCU register name and each register retention value are displayed.

This function (command) can be used only when SOFTUNE Workbench is in the debug session.

Selecting [Register] when the Register Window has been already opened activates the Window.

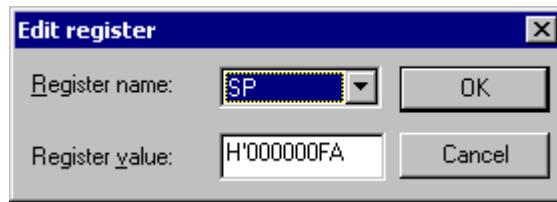
■ Changing Register Values

The values saved by the registers displayed in the Register Window can be changed directly by the following procedures:

- Full change
 1. Double-click a register name or display value.
 - The register name or display value is reversely displayed.
 2. Specify the values to set in turn using a hexadecimal number, starting from the highest-order digit (leftmost digit).
 - The register value is set automatically when the digit in the lowest-order bit (rightmost digit) is changed.
- Partial digital (bit field) change
 1. Click the digit to be changed in the register value display.
 - The cursor appears at the left of the clicked digit.
 2. Set the new value as a 1-digit hexadecimal number.
 - The cursor automatically moves to the right digit. If the changed digit is the lowest-order digit, the register value is updated automatically.
 3. When changing of the required number of digits is completed, click another register name or register value display.
 - If the Register Window is closed without clicking another register name or register value display, the changed value is not set in the register.
 - This operation is not necessary when the change to the lowest-order bit is completed.
- Selected register name change
 1. Click the right button of the mouse in the Register Window to display the short-cut menu, then select [Edit].

The register edit dialog shown in Figure 4.4-3 opens.
 2. Select a register name.
 3. Input the value to set.
 4. Click the [OK] button.

Figure 4.4-3 Resister Edit Dialog



- Register name
Specify a register name.
- Register value
Specify the value set in a register

4.4.5 Memory

"Memory" displays the Memory Window.

■ Displaying Memory Data

When the Memory Window is opened, memory data dump is displayed, starting from the specified address. Memory data can be also modified using the short-cut menu.

This function (command) can be used only when SOFTUNE Workbench is in the debug session.

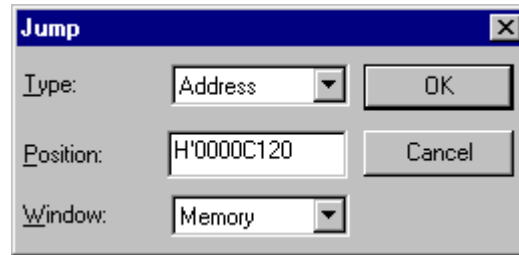
- When the Memory Window has been already opened

The Memory Window is activated.

- When the Memory Window is not opened.

The dialog for specifying the display start address (Figure 4.4-4) opens.

Figure 4.4-4 Dialog for Specifying Display Start Address



- Type

Specify the type of display start position (Line number/Address/Frame).

- Position

Display disassembling from the position specified in the above type.

- Window

To specify the window where jump is implemented. For details, see Section "4.3.6 Jump".

If it is a disassemble window, default will be shown as [Memory].

■ Changing Memory Data

Memory data can be changed by rewriting the displayed dump value directly. When a character string is entered from the ASCII character string display field, the ASCII code of each character is set automatically in the corresponding address.

The address where the memory dump is to be started can be changed by changing the address field display.

■ Add Bookmark

Please click on the right button of the mouse on memory window to display short-cut menu. Then, select [Add bookmark]. [Add bookmark] Dialog will be displayed. (Figure 4.4-5, Figure 4.4-6)

Figure 4.4-5 Add Bookmark Dialog (Address)

The screenshot shows the 'Add bookmark' dialog box with the following fields and options:

- Bookmark name:** H'C313
- Color:** A color selection dropdown menu.
- Area:** A group box containing two radio buttons: ☒ Address and ☐ Symbol.
- Start address:** H'C313
- End address:** H'C316
- Buttons:** OK and Cancel.

Figure 4.4-6 Add Bookmark Dialog (Symbol)

The screenshot shows the 'Add bookmark' dialog box with the following fields and options:

- Bookmark name:** target[0]
- Color:** A color selection dropdown menu showing red.
- Area:** A group box containing two radio buttons: ☐ Address and ☒ Symbol.
- Variable name:** target[0]
- Buttons:** OK and Cancel.

● Bookmark name

Specify the name of bookmark. Default will be changed as follows depending on setup made in [Area].

If the [Area] is [Address]: Start address

If the [Area] is [Symbol]: Variable name

● Color

Specify the color of background in bookmark.

- Area

Specify the method for determining the address range of bookmark.

Address: Specify with start address or end address

Symbol: Specify the address range with the name of symbol

- Start address

Specify the start address of bookmark.

- End address

Specify the end address of bookmark.

- Variable name

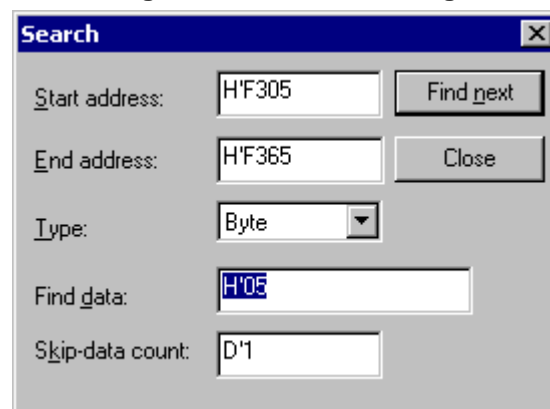
Specify the name of symbol for setting a bookmark.

■ Searching Memory Data

Click the right button of the mouse in the Memory Window to display the short-cut menu, then select [Search] from the menu.

The search dialog shown in Figure 4.4-7 opens.

Figure 4.4-7 Search Dialog



- Start Address

Specifies start address of search range.

- End Address

Specifies end address of search range.

- Type

Selects data type (Byte/Word/Long/ASCII).

- Find Data

Specifies matching data type.

When ASCII is selected as the data type, specify a character string. To search for several data items continuously when the data type is not ASCII, write each item, delimited by a comma (,).

- Skip Data Count

To search the search range continuously, set 1. When a value greater than or equal to 2 is set as the skip byte count, addresses are skipped for each set count and the search range is searched.

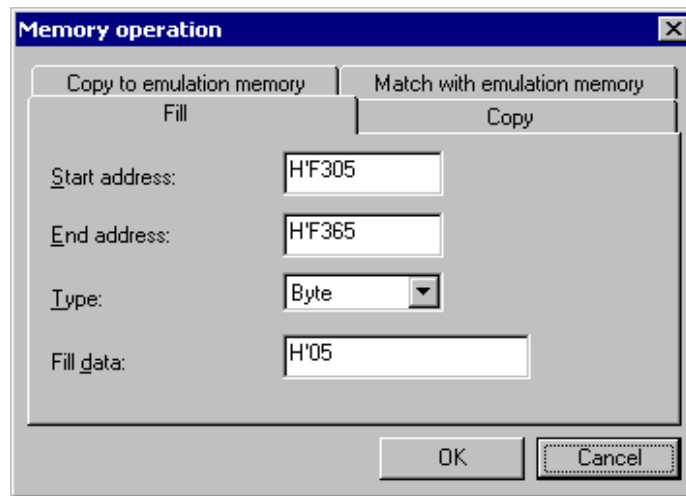
■ Special Operation

To fill memory with data or to copy data to memory, start debugging, click the right button of the mouse in the Memory Window to display the menu, then select [Special Operation] from the menu. The memory operation dialog opens.

Clicking the [OK] button from this dialog starts the function of the open tab.

-Fill

Figure 4.4-8 Memory Operation Dialog (Fill)



To fill memory with data, open the [Fill] tag, then set the start address, end address, data type, and filling data.

-Start Address

Specifies start address of memory area to be filled with data. Data filling is started at this address.

-End Address

Specifies end address of memory area to be filled with data. Data filling is continued to this address.

-Type

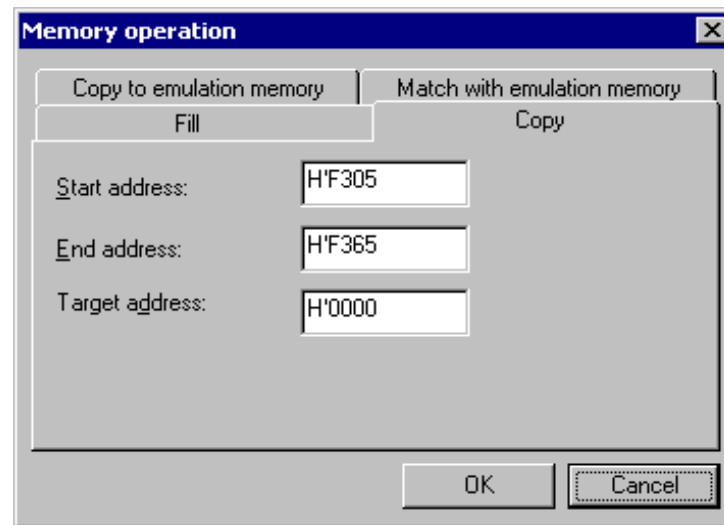
Specifies type of filled data. (Byte/Word/Long/ASCII).

-Fill Data

Specifies filling data to fill specified memory area. Several filling data can be specified, delimited by a comma (,).

-Copy

Figure 4.4-9 Memory Operation Dialog (Copy)



To copy data to memory, open the [Copy] tag, then set the copy source start address, copy source end address, and copy destination start address.

-Start Address

Specifies start address of copy source area. Data copy is started at this address.

-End Address

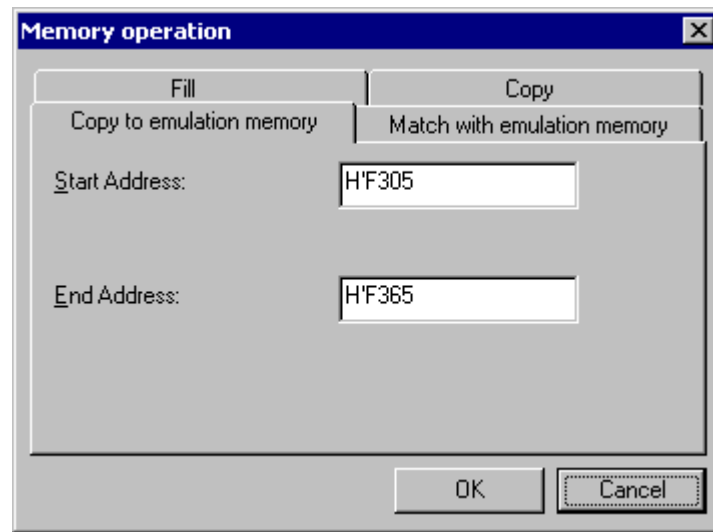
Specifies end address of copy source area. Data copy is continued to this address.

-Target Address

Specifies start address of copy destination area.

-Copy to emulation memory (MB2141)

Figure 4.4-10 Memory Operation Dialog (Copy to Emulation Memory)



To copy data from user memory to emulation memory, open the [Copy to Emulation Memory] tag, then set the transfer start address and transfer end address.

-Start Address

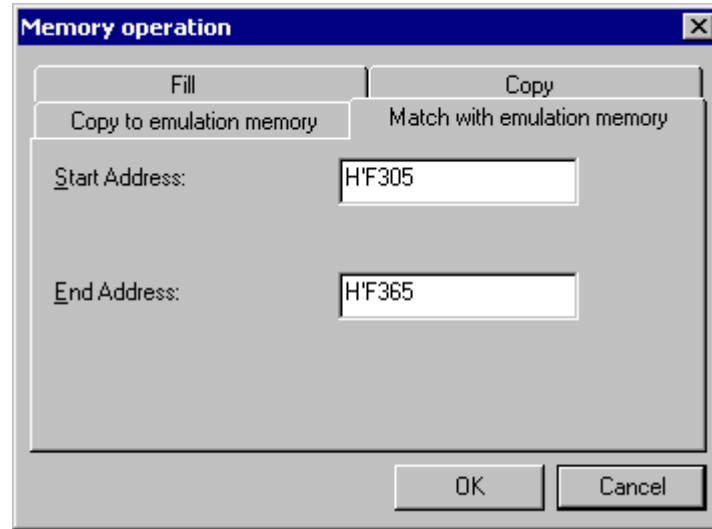
Specifies address at which data transfer is to be started. Copy to emulation memory is started at this address.

-End Address

Specifies end address of transfer source area. Copy to emulation memory is continued to this address.

-Match with emulation memory (MB2141)

Figure 4.4-11 Memory Operation Dialog (Match with Emulation Memory)



To collate user memory with emulation memory, open the [Match with Emulation Memory] tag, then set the collocation start address and collation end address.

-Start Address

Specify the address at which collation is to be started. Collating user memory with emulation memory is started at this address.

-End Address

Specify the end address of the collation source area. Collating user memory with emulation memory is continued to this address.

■ Comparing Memory Blocks

To compare memory blocks, click the right button of the mouse in the Memory Window to display the menu, then select [Compare] from the menu. The comparison dialog shown in Figure 4.4-12 opens.

-Start Address

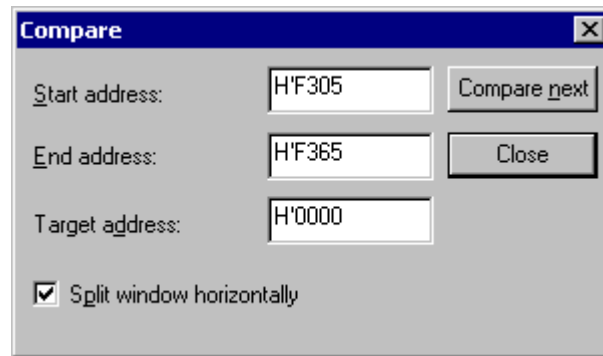
Specifies start address of comparison source area. Memory block comparison is started from this address.

-End Address

Specifies end address of comparison source area. Memory block comparison is continued up to this address.

-Target Address

Specifies start address of comparison destination area.

Figure 4.4-12 Comparison Dialog

For example, when 4 is specified as the skip byte count, addresses are skipped for each 4 bytes like address 4 and address 8 and the search range is searched.

[Example]

Suppose the data in memory is "0000 01 02 03 01 02 01 02 03 04 01 01 02 03".

When Data Type is Byte, Search Data is 01, and Skip Byte Count is 1, all search data is found. However, when Skip Byte Count is 2, only 01 search data at address 0000 and 000A is found. 01 search data at other addresses is not found.

When Data Type is Byte, Search Data is 01 and 02, and Skip Byte Count is 3, only 01 and 02 search data at address 0000 and address 0003 is found. 01 and 02 search data at address 0005 and address 000A is not found.

■ Edit

To edit memory data, click the right button of the mouse in the Memory Window to display the menu, then select [Edit] from the menu. The edit dialog shown in Figure 4.4-13 opens.

-Address

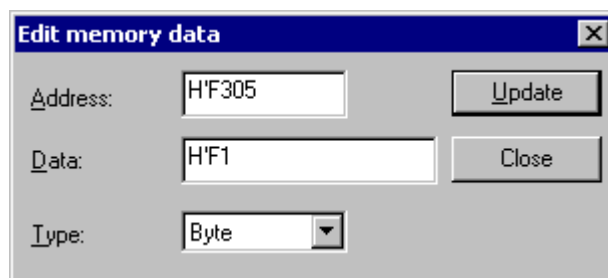
Specifies address to edit.

-Data

Specifies memory data to rewrite.

-Type

Specifies size of data to rewrite. (Byte/Word/Long)

Figure 4.4-13 Memory Edit Dialog

■ Setup

To set the display format of the Memory Window, click the right button of the mouse in the Memory Window, then select [Setup] from the menu. The setup dialog shown in Figure 4.4-14 opens.

- Type

Specifies display format of Memory Window.

"Bit", "byte", "word", or "long" can be selected.

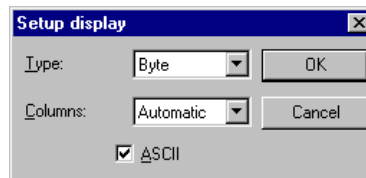
- Type

Selects whether to display ASCII characters at right of Memory Window.

- Columns

To specify the number of bytes display in a line. The number of bytes can be chosen from Automatic, 4byte, 8byte, 16byte, 32byte or 64byte.

Figure 4.4-14 Setup Dialog



4.4.6 Local

"Local" displays the Local Variable Window.

■ Local Variable Window

The Local Variable Window displays, in tree format, the local variables of the function where the current instruction pointer exists using the function name as the root. The displayed variables cannot be added nor the displayed variables can be canceled.

The variable values modified as a result of program execution are updated automatically. For this reason, the user can observe how variable values change as a result of program execution.

Variable values can be also modified to continue debugging.

■ Setting Radix

The Radix when a variable value is displayed can be set for each variable using the following procedure:

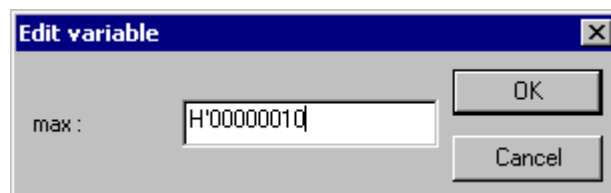
1. Click the right button of the mouse in the line containing the variable of the radix to be modified.
The short-cut menu is displayed.
2. Move the mouse cursor to [Radix].
The list showing selectable radices is displayed in the submenu.
3. Select the radix to modify from the list.

■ Modifying a Variable Value

The variable value can be modified by the following procedure:

1. Click the right button of the mouse in the line containing the variable to be modified.
The short-cut menu is displayed.
2. Select [Edit] from the short-cut menu.
The variable edit dialog shown in Figure 4.4-15 opens.
3. Set a variable value, then click the [OK] button.

Figure 4.4-15 Variable Edit Dialog

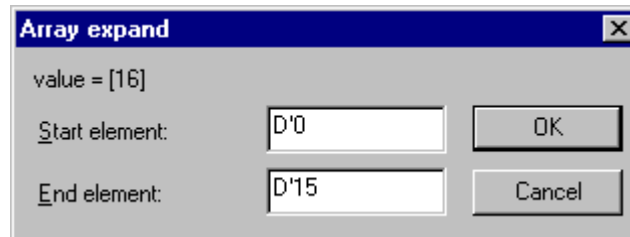


■ Setting Display Range at Array Expansion

When an array is expanded, a range of the element displayed in the window can be specified.

When clicking the right button of the mouse on an array variable in the local variable window to select [Element] in the short-cut menu, the array expansion dialog (Figure 4.4-16) is displayed.

Figure 4.4-16 Array Expansion Dialog



- Start element number

Specifies start element number displayed at array expansion.

- End element number

Specifies end element number displayed at array expansion.

Only the specified range is displayed at array expansion. At default, all elements are displayed.

4.4.7 Watch

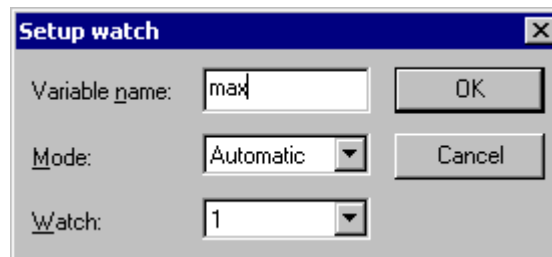
"Watch" displays the Watch Window.

■ Watch Window

The Watch Window displays the values of the specified variables in tree format. The variable values modified as a result of program execution are updated automatically. For this reason, the user can observe how variable values change as a result of program execution. The short-cut menu can be used to modify the displayed variable values.

1. Click the right button of the mouse in the Watch Window.
The short-cut menu is displayed.
2. Click [Setup].
The watch setup dialog shown in Figure 4.4-17 opens.
3. Input a variable name from the dialog. Also select a [mode] as required.
4. Select a [watch] window number.
5. Click the [OK] button.

Figure 4.4-17 Watch Setup Dialog



- Variable name
Specify the name of a variable to be displayed.
- Mode
Specify the mode, C or assembler language, in which a variable is displayed. For automatic operation, the variable is displayed in the predetermined language mode (Automatic/ C language/Assembler).
- Watch
Specify the Watch Window Number name.

■ Setting Radix

The Radix when a variable value is displayed can be set for each variable using the following procedure:

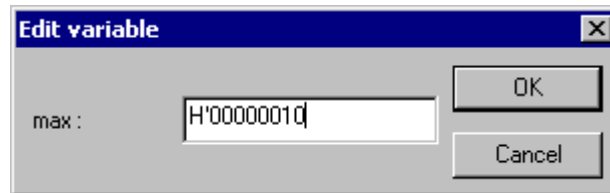
1. Click the right button of the mouse in the line containing the variable of the radix to modify.
The short-cut menu is displayed.
2. Move the mouse cursor to [Radix].
The list showing selectable radices is displayed in the submenu.
3. Select the radix to modify from the list.

■ Modifying a Variable Value

The variable value can be modified using the following procedure:

1. Click the right button of the mouse in the line containing the variable to modify.
The short-cut menu is displayed.
2. Select [Edit] from the short-cut menu.
The variable edit dialog shown in Figure 4.4-18 opens.
3. Set a variable value, then click the [OK] button.

Figure 4.4-18 Variable Edit Dialog



■ Canceling Display of Unnecessary Variable

Display of a set variable that is no longer needed can be canceled from the Watch Window using the following procedure:

1. Click the right button of the mouse in the line displaying the variable to cancel.
The short-cut menu is displayed.
2. Select [Delete] from the short-cut menu.

Note:

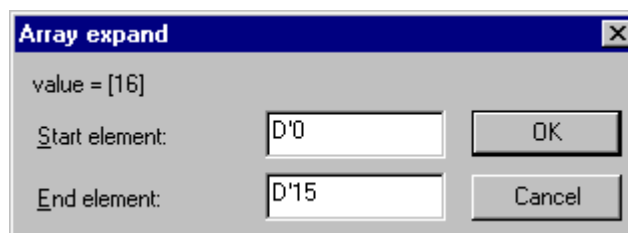
Display of each array element or each member (e.g., structure) cannot be canceled. When [delete] is executed in each element or member line, display of the array or structure is canceled.

■ Setting Display Range at Array Expansion

When an array is expanded, a range of the element displayed in the window can be specified.

When clicking the right button of the mouse on an array variable in the watch window to select [Element] in the short-cut menu, the array expansion dialog (Figure 4.4-19) is displayed.

Figure 4.4-19 Array Expansion Dialog



- Start element number

Specifies start element number displayed at array expansion.

- End element number

Specifies end element number displayed at array expansion.

Only specified range is displayed at array expansion. At default, all elements are displayed.

In the watch window, the setting value is restored when reactivating the debugger.

Note:

The value of one-dimensional array registered as the watch variable is only restored.

4.4.8 Trace

"Trace" displays the Trace Window.

■ Trace

This function retroactively displays addresses and instructions executed so far. [Enable] can be switched to [Disable] or vice versa during debugging. Display per machine instruction, cycle display, display per source, or display per task (used by REALOS only) can be selected as trace result display.

■ Refresh

The Trace Window display is not refreshed in real-time according as debugging progresses. Consequently, to display the latest trace, click the right button of the mouse in the Trace Window to display the short-cut menu, then select [Refresh] from the menu.

■ Buffer Size

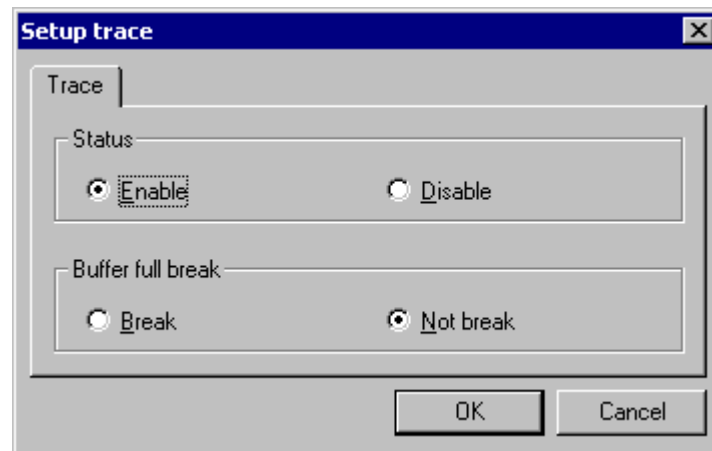
Trace data is buffered in the trace buffer. The trace buffer becomes full some time during debugging because its size is finite. When the trace buffer becomes full, the program being executed can be stopped.

■ Trace Setup

Select [Setup] from the short-cut menu.

-Trace [Simulator/MB2141]

Figure 4.4-20 Trace Setup Dialog



-Status

Specifies control status.

-Buffer Full Break

Specifies whether or not to break trace buffer full.

Note:

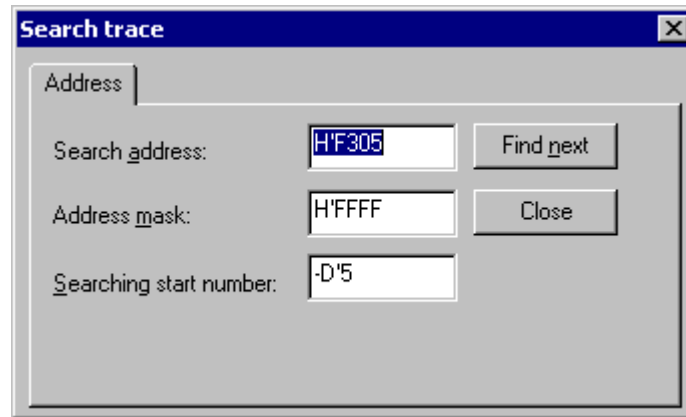
In single trace and multitrace, items are individually set.

■ Trace Search

Select [Search] from the short-cut menu.

-Address (Simulator)

Figure 4.4-21 Trace Search Dialog (Address)



-Search address

Specifies start address of search range.

-Address mask

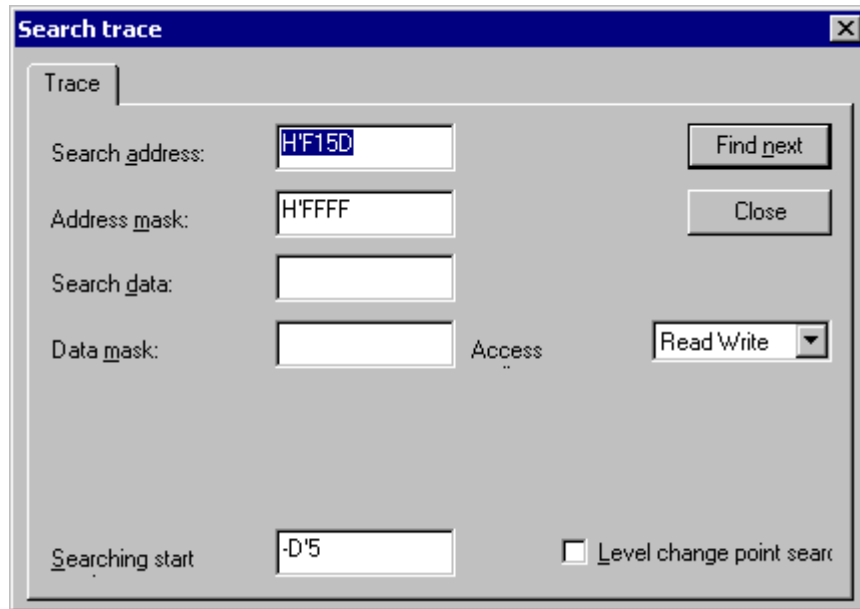
Specifies mask address of search range.

-Searching start number

Specifies number of frame where search to start.

-Trace (Emulator)

Figure 4.4-22 Trace Search Dialog (Trace)



-Search Address

Specifies address to search.

-Address Mask

Specifies address mask. Only bit-1 addresses are compared.

-Search Data (MB2141)

Specifies data to find.

-Data Mask (MB2141)

Compares only 1-bit data.

-Search Start Number

Specifies number of frame where search to start.

-Access Attribute (MB2141)

Specifies access attribute. (Code/Read/Write/Code Read/Read Write/Code Read Write)

-Level Change Point Search (MB2141)

Specifies sequencer level change point. This item is not displayed when multitrace search is executed.

■ Example of address Mask

The relationship among search address, address mask and the actual address to be found is shown in Table 4.4-1.

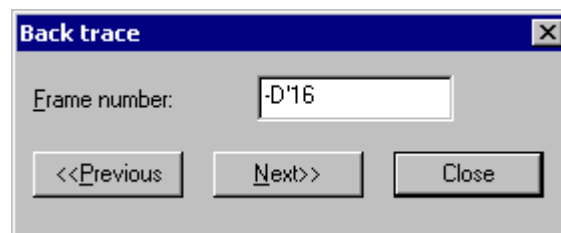
Table 4.4-1 Relationship Among Search Address, Address Mask, and the Actual Address to be Found

Search address	Address mask	Actual address to be found
H'F8CA	H'FFFF	H'F8CA
	H'FF00	Any address from H'F800 to H'F8FF

■ Back Trace

Select [Back Trace] from the short-cut menu.

Figure 4.4-23 Back Trace Dialog



- Frame Number

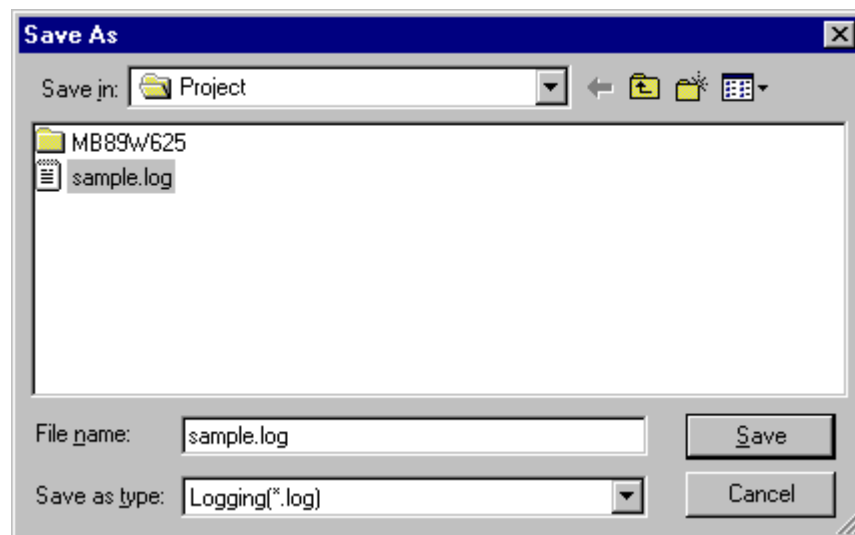
Specifies frame number corresponding to Trace Window.

■ Save

Select [Save] from the short-cut menu.

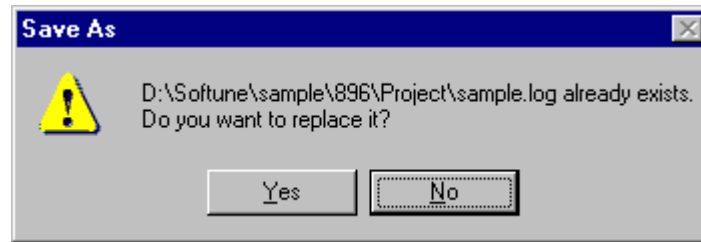
Save the trace data to the specified file.

Figure 4.4-24 Save Trace Data Dialog



When specifying a file name, specify an existing file, and select whether to save or not.

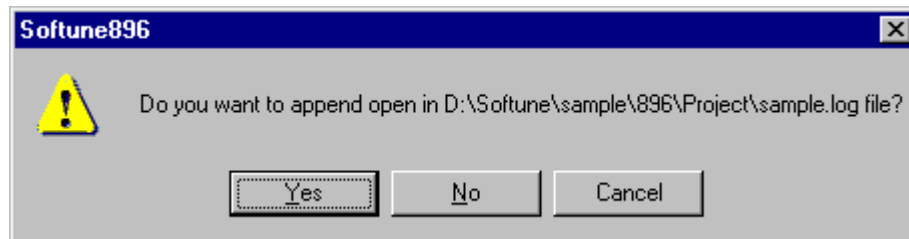
Figure 4.4-25 Save Dialog



Select whether to add and save when selecting "Yes" (Save).

Selecting "No" does not save, trace data to the file.

Figure 4.4-26 Add/Save Dialog



Selecting "Yes" (Add/Save) adds and saves trace data to the file.

Selecting "No" saves trace data to in the file.

4.4.9 Command

"Command" displays the Command Window.

■ Command

A debugger command can be entered and executed directly from the displayed Command Window. The command execution result is also displayed in the Command Window. For the supported debugger commands, refer to "SOFTUNE Workbench Command Reference Manual".

4.4.10 Tool Bar, Status Bar, Tab

Tool bar sets display items. Status Bar or Tab switches display to nondisplay or vice versa.

■ Tool Bar

Any of the following tool button sets displayed in the tool bar can be selected:

- Common bar
- Find
- Debug
- Flag
- Flash
- Build
- Project

For the buttons included in the above sets, see Section "3.2 Tool Bar".

■ Status Bar

"Status Bar" can only switch display to nondisplay or vice versa. For status bar display items, see Section "3.3 Status Bar".

■ Tab

This function enables to choose whether the tab is displayed or non-displayed for switching. By selecting tab function, tab will be attached to the windows, Thus, this makes easy to switch windows.

4.4.11 Object

"Object" displays the Object Window.

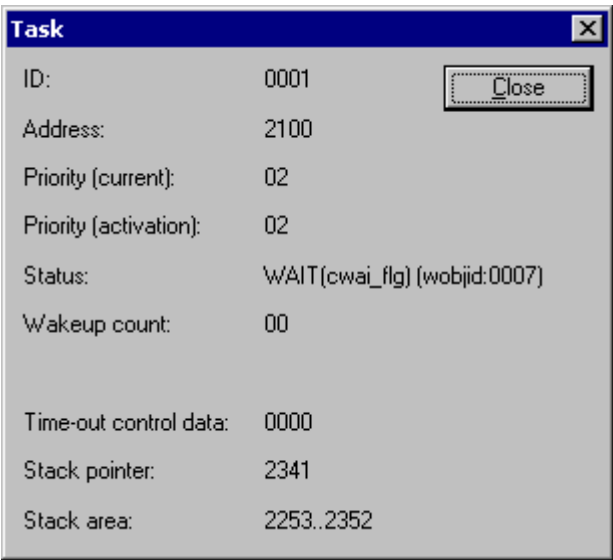
■ Object Window

The Object Window displays the detailed object information used by REALOS. This window can be also used to issue system calls.

To display detailed object information, select an ID, then select [Property] from the short-cut menu.

■ Detailed Display of Task

Figure 4.4-27 Detailed Task Display Dialog

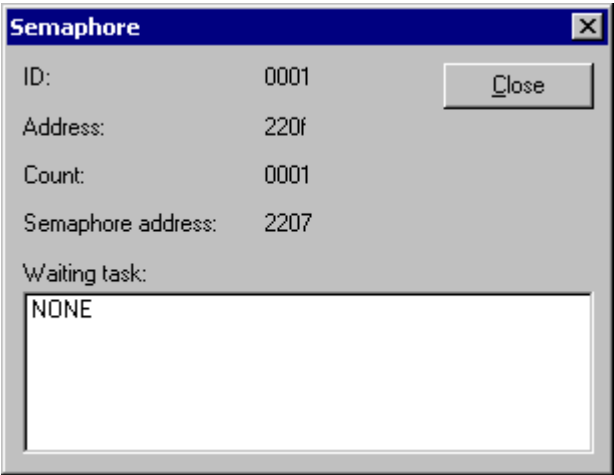


- | | |
|-------------------------|--|
| - ID | Task ID number |
| - Address | Task control block address |
| - Priority (Current) | Current task priority |
| - Priority (activation) | Task priority at start-up |
| - Status | Task state |
| TTS_RDY | Running or ready state |
| TTS_DMT | Dormant state |
| TTS_WAI | Wait state |
| TSS_SUS | Suspend state |
| Wait factor: | The wait factor is displayed when SOFTUNE Workbench is in the object wait state. |
| TTW_SLP | slp_tsk/tslp_tsk wait state |
| TTW_DLY | dly_tsk wait state |
| TTW_SEM | wai_sem wait state |

TTW_FLG	wai_flg wait state
TTW_MBX	rcv_msg wait state
TTW_MPL	get_blk wait state
- Wakeup Count	Wakeup request count
- Timeout Control Data	Timeout management data
- Stack Pointer	Stack pointer
- Stack Area	Stack area

■ Detailed Display of Semaphore

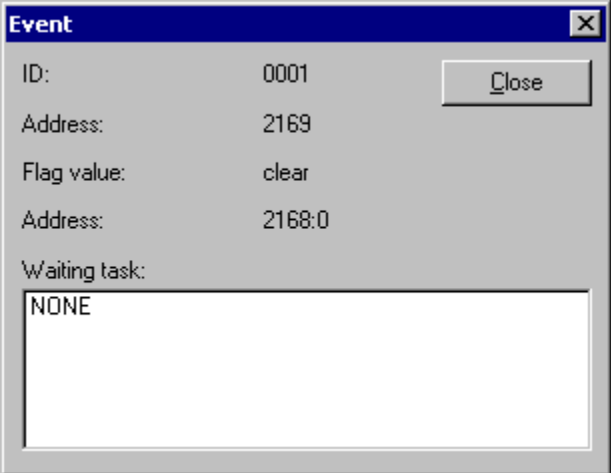
Figure 4.4-28 Detailed Semaphore Display Dialog



- ID	Semaphore ID number
- Address	Address of semaphore control block
- Count	Current semaphore count
- Semaphore Address	Semaphore queue address
- Waiting Task	ID number of task in queue

■ Detailed Display of Event Flag

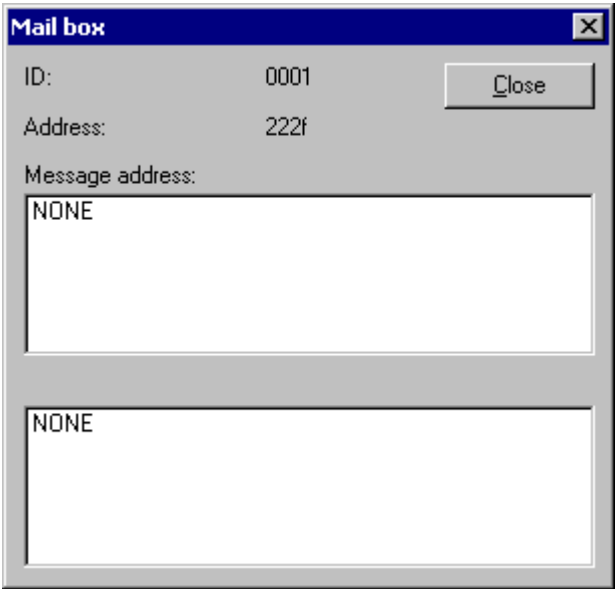
Figure 4.4-29 Detailed Event Flag Display Dialog



- ID Event flag ID number
- Address Address of event flag control block
- Flag value Current event flag pattern
- Flag address Address of current event flag table
- Waiting Task ID number of task in queue

■ Detailed Display of Mailbox

Figure 4.4-30 Detailed Mailbox Display Dialog

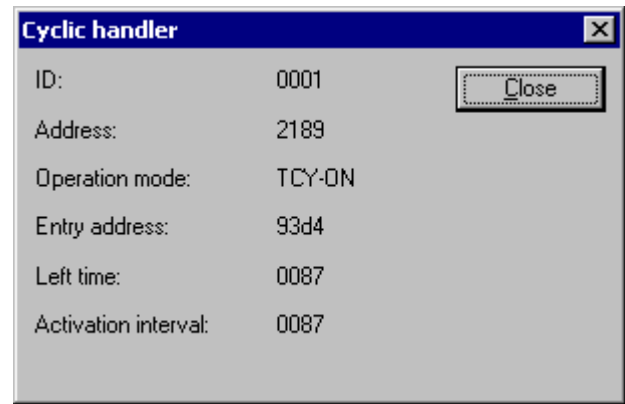


- ID Mailbox ID number
- Address Address of mailbox control block
- Message State Message address
- Waiting Task ID number of task in queue

Waiting Task or Message State is displayed according to the wait factor.

■ Detailed Display of Cyclic Handler

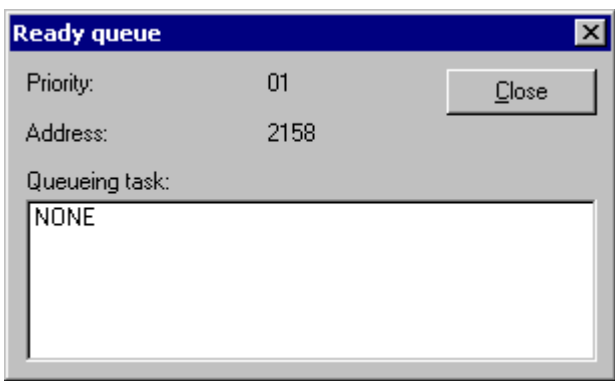
Figure 4.4-31 Detailed Cyclic Handler Display Dialog



- ID ID number of cyclic handler
- Address Address of cyclic handler control block
- Operation Mode Cyclic handler state (Enable/Disable)
- Entry Address Cyclic handler address
- Left Time Residual time till activation
- Activation Interval Cyclic time interval

■ Detailed Display of Ready Queue

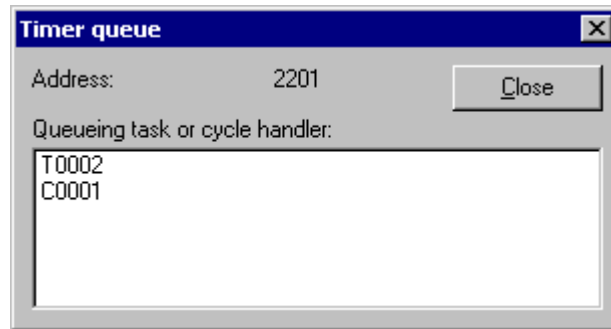
Figure 4.4-32 Detailed Ready Queue Display Dialog



- Priority Priority
- Address Address of ready queue header table
- Queueing task ID number of queued task

■ Detailed Display of Timer Queue

Figure 4.4-33 Detailed Timer Queue Display Dialog



- Address Timer queue address
- Queueing task or cycle handler Cyclic handler ID placed in the timer queue or timeout waiting task ID

4.4.12 Coverage

"Coverage" displays the Coverage Window.

■ Coverage Window

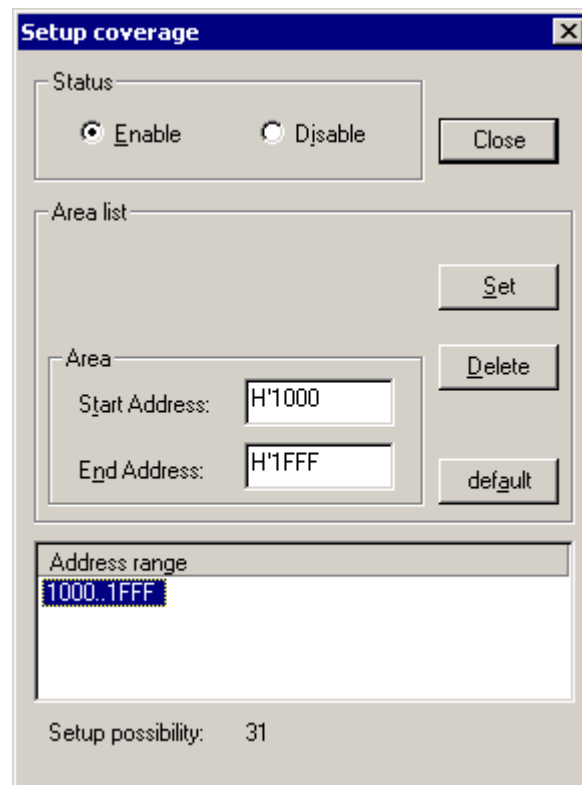
The window displaying the coverage measurement result is opened for setting the coverage and displaying the total coverage rate.

This window can be used only for the MB2141 (emulator).

■ Coverage Setup

Select [Setup] from the short-cut menu of the Coverage Window.

Figure 4.4-34 Coverage Setup Dialog



- Status Select [Enable] or [Disable].

- Area list Set [Attribute] and [Area].

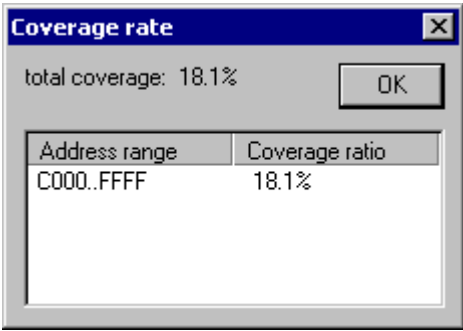
To delete an area, click the [Delete] button.

To set an area automatically after the load module file has been read, click [Default].

■ Coverage Rate

Select [Total Coverage] from the short-cut menu of the Coverage Window.
The total coverage rate and the coverage rate for each address range are displayed.

Figure 4.4-35 Coverage Rate Dialog



4.4.13 Performance

"Performance" displays the Performance Window.

■ Performance Window

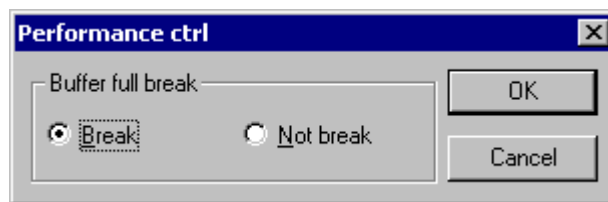
Open the Performance Window to display the performance measurement result. The Performance Window is used to control and set performance.

This function can be used only in the MB2141 (emulator).

■ Performance Control

Select [Setup] from the short-cut menu of the Performance Window. When the buffer is full, set whether to break buffer full.

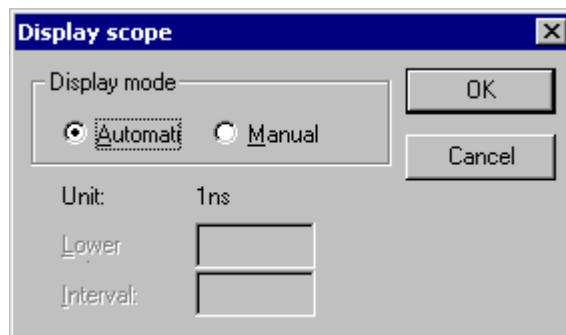
Figure 4.4-36 Performance Control Dialog



■ Performance Setup

Select [Display Scope] from the short-cut menu of the Performance Window.

Figure 4.4-37 Performance Setup Dialog



- Display Mode

Select [Automatic] or [Manual]. When [Manual] is selected, set [Lower Scope] and [Interval].

- Lower

Specify display start time for detailed measurement result display.

- Interval

Specify display interval time for detailed measurement result display.

- Unit

The values set in [Timer minimum measuring unit] of the [Emulation] tab in the debug environment setup dialog are displayed. This dialog opens when [Debug environment] is selected from the [Setup] menu.

■ Copy of the Performance Measurement Result

The measurement result displayed in the performance window can be copied to a clipboard. To do this, select [Copy] from the short-cut menu while the measurement result is displayed in the performance window.

[Examples]

When the display type of MB2141 is "measurement time"

event = 1 -> 2
 min time = 1158.0
 max time = 1830.0
 avr time = 1503.0

time (μs)		count
0.0 -	999.0	0
1000.0 -	1099.0	0
1100.0 -	1199.0	1
1200.0 -	1299.0	3
1300.0 -	1399.0	0
1400.0 -	1499.0	0
1500.0 -	1599.0	0
1600.0 -	1699.0	0
1700.0 -	1799.0	2
1800.0 -	1899.0	2
1900.0 -	1999.0	0
2000.0 -		0
total		8

When the display type of MB2141 is "measurement count"

no.	count
1	9
2	8
5	17
6	1
7	1
8	1

Note:

The contents output to a clipboard by [Copy] form the short-cut menu are same as those output by SHOW PERFORMANCE command. For more information, please refer to Section "4.8 SHOW PERFORMANCE" in SOFTUNE Workbench Command Reference Manual.

4.4.14 Fonts

The Fonts for each window are changed.

■ Setting Fonts

The font information (font name and size) currently set for each window can be displayed and the setting can be changed. Also, all font settings can be set to the defaults.

During debugging, the fonts for debug-related windows (such as Source window) can be changed.

■ Changing Fonts

Change fonts as follows:

1. Select the [View]-[Font] menu.
 - The font setting dialog is displayed (Figure 4.4-38).
2. Select the window with the font to be changed and click the [Font...] button.
 - The font setting dialog is displayed (Figure 4.4-38).
3. Specify the font name and size and click the [OK] button. The font type that can be selected depends on the window.
4. When the [OK] button in the font setting dialog (Figure 4.4-38) is clicked, the window fonts are changed.

■ Resetting Fonts

Reset fonts as follows:

1. Select the [View]-[Font] menu.
 - The font setting dialog is displayed (Figure 4.4-38).
2. Click the [All Reset] button.
3. The fonts for the window displayed in [Window] are all reset to the defaults.
4. When the [OK] button is clicked, the window fonts are returned to the defaults.

Figure 4.4-38 Set Font Dialog

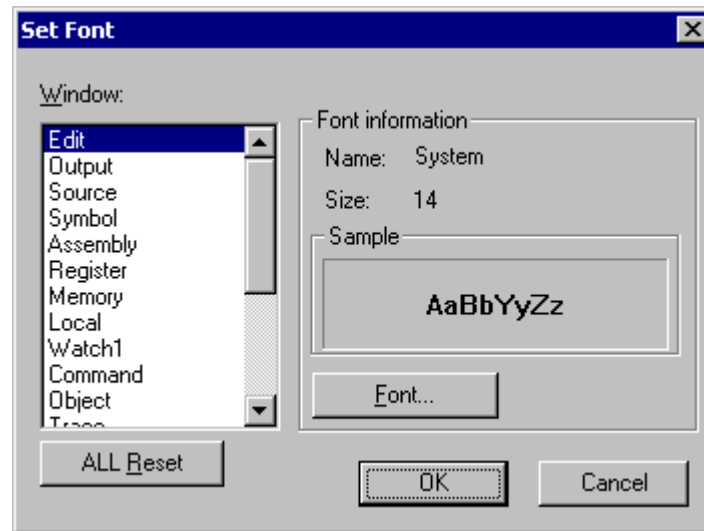
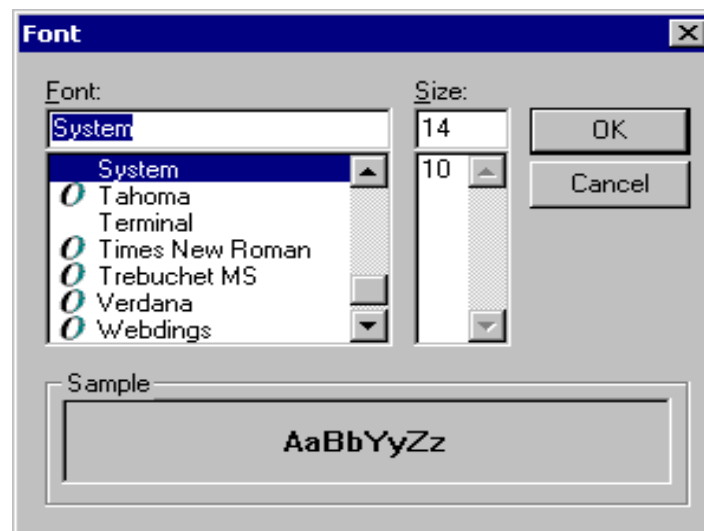


Figure 4.4-39 Font Dialog



4.5 Project

"Project" sets project-related items and executes make/build.

■ Project-related Item Setup

- Active Project
- Add Project
- Add Member
- Setup Workspace
- Setup Project
- Setting Customize Build
- Project Dependencies
- Configuration
- Include Dependencies

■ Make/build Execution

- Compile
- Make
- Build
- Stop

Note:

No selection can be made during debug session. Quit the debugger.

4.5.1 Active Project

The active project is replaced.

■ Active Project

When the names of all projects in workspace are displayed in the submenu, click the name of the project that is made active.

A checkmark is placed at the left side of the active project name in the menu.

4.5.2 Add Project

A project is added to workspace.

■ Add Project

There are the following two menus to add a project:

- New

A new project is created and stored in workspace.

- Project

An existing project is stored in workspace.

Note:

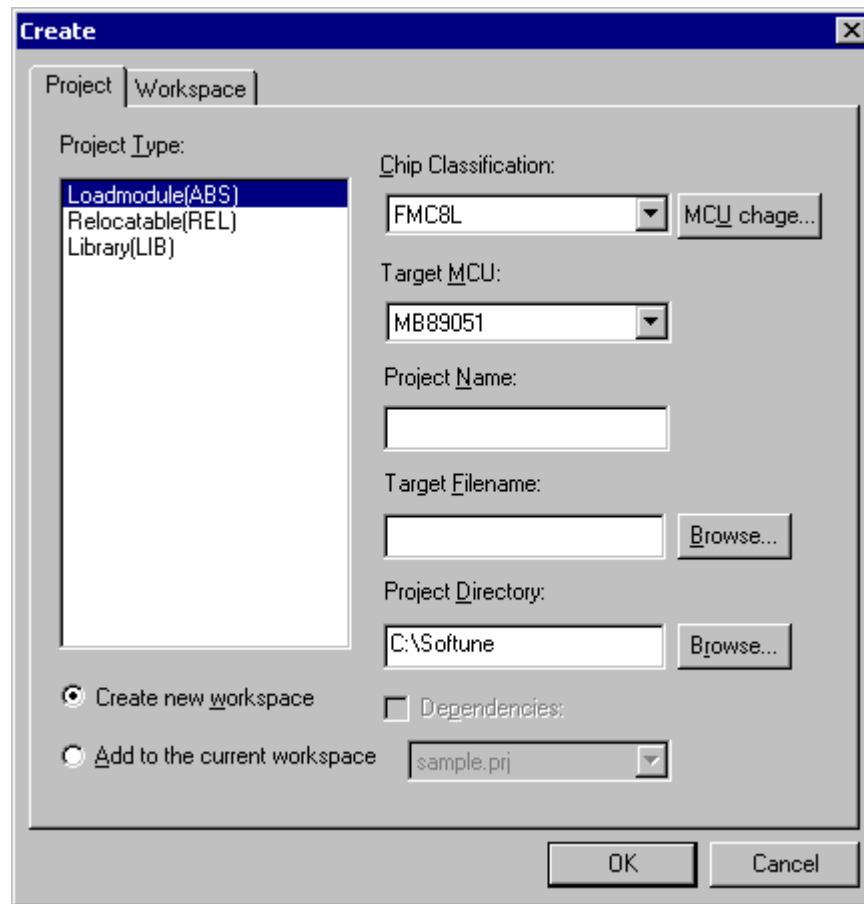
If any project having the same name as that of the project to be stored is in workspace, it cannot be stored in workspace.

4.5.2.1 Add Project - Create

A new project is created and stored in workspace.

■ Add Project - Create

Figure 4.5-1 Create Project Dialog



- Project Type
Select the project type. For details about the project type, see Section "2.4 Storing of Project".
- Create new workspace
New workspace is created for storage of a project. This item cannot be selected.
- Add to the current workspace
A project is stored in the currently opened workspace.
- Chip Classification
Selects the chip classification.
- Target MCU
Selects the target MCU.

- Project Name
 - Specifies the project name.
- Target File name
 - Specifies the target file name.
- Project Directory
 - Specifies the directory of the project.
- Dependencies
 - When the project name is specified in the combo box, the project to be stored is defined as a subproject in the specified project.
- MCU change
 - Opens MCU change dialog.

■ Procedure for Addition

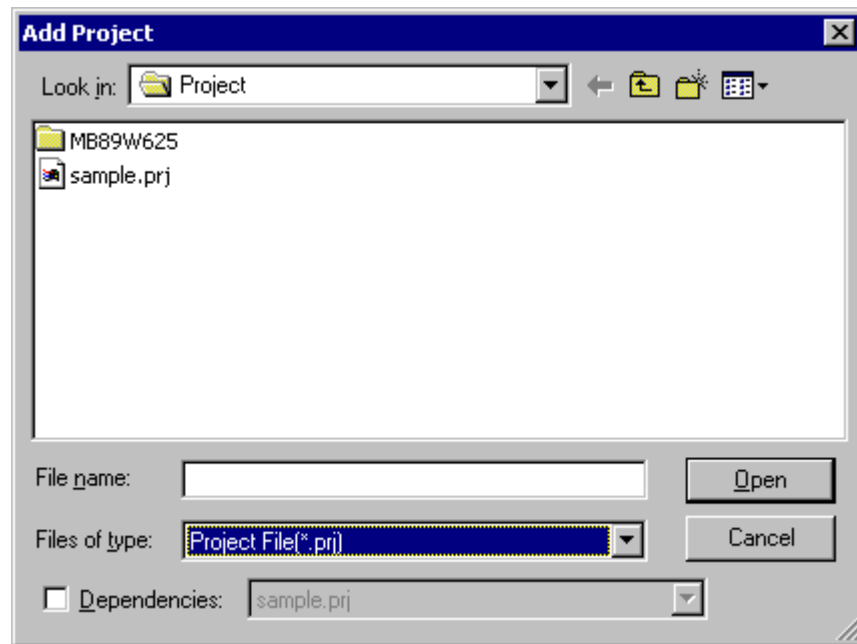
For the procedure for addition of a project, see Section "2.4 Storing of Project".

4.5.2.2 Add Project - Existing Project

An existing project is stored in workspace.

■ Add Project - Existing Project

Figure 4.5-2 Add Project Dialog



- Dependencies

When the project name is specified in the combo box, the project to be stored is defined as a subproject in the specified project.

■ Procedure for Addition

For the procedure for addition of a project, see Section "2.4 Storing of Project".

4.5.3 Add Member

"Add Member" adds a file to the project.

■ Add Member

There are the following menus to add a member:

- File

A file is specified and stored in the project.

- Directory

A directory is specified to store its file and folder in the project.

The file dialog box for file selection is displayed. Two or more files can be selected at a time.

The selected files are added to the project and displayed in the Project Window.

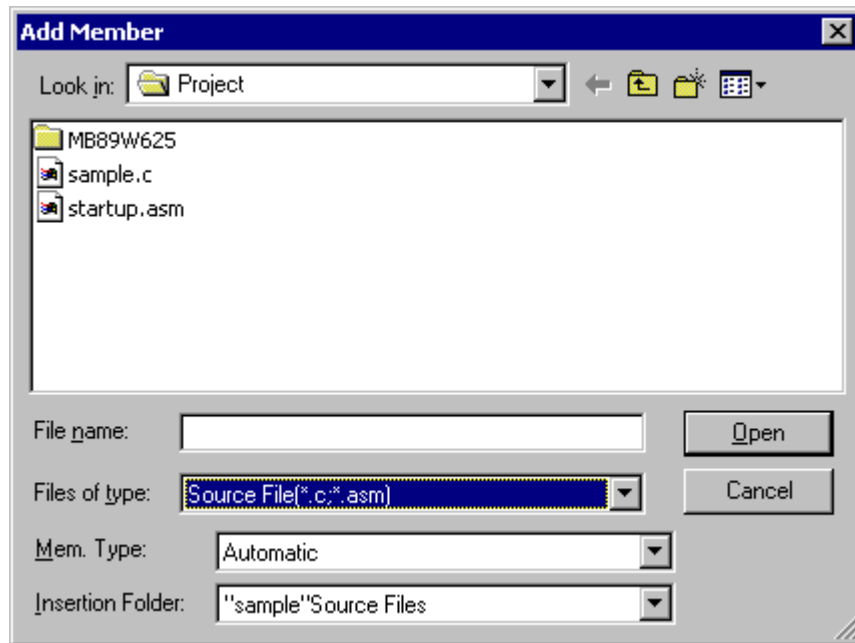
4.5.3.1 Add Member - File

A file is specified and stored in the project.

■ Add Member - File

The file dialog box for file selection is displayed. Two or more files can be selected at a time.

Figure 4.5-3 Add Member Dialog-File



- Mem. Type

The file to be stored is stored as the type of specified member in the project. If "Automatic" is selected, the type of member is determined by the extension.

- Insertion Folder

The project in which a file is stored and the folder into which the file is inserted are specified. The character string enclosed between "and" is the project name.

■ Procedure for Addition

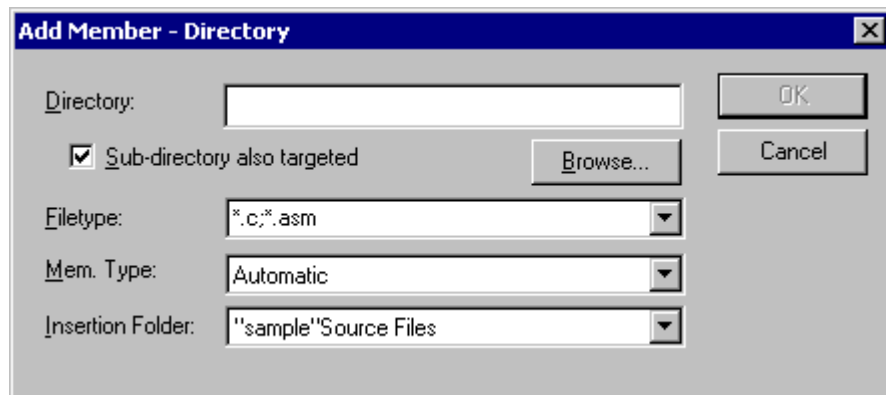
For the procedure for addition of a member, see Section "2.5 Creating and Registering Source File in Project".

4.5.3.2 Add Member - Directory

A directory is specified to store its file and folder in the project.

■ Add Member - Directory

Figure 4.5-4 Add Member Dialog-Directory



- Directory

Specifies the directory having the file to be stored.

- Sub-directory also targeted

When turned on, the file in the sub-directory in the specified directory is stored. A sub-directory is hierarchically created as a folder in the project.

- File type

Only the file having a specified extension is stored in the project.

- Mem. Type

The file is stored as the type of the specified member in the project. If "Automatic" is selected, the type of member is determined by the extension.

- Insertion Folder

Specify the project in which a file is stored and the folder into which the file is inserted. The character string enclosed between "and" is the project name.

■ Procedure for Addition

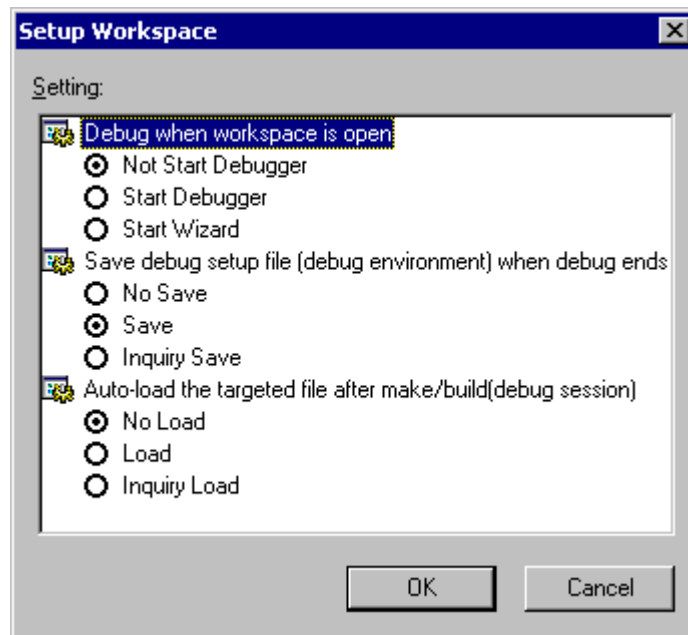
For the procedure for addition of a member, see Section "2.5 Creating and Registering Source File in Project".

4.5.4 Setup Workspace

The basic setting regarding workspace is performed.

■ Setup Workspace

Figure 4.5-5 Setup Workspace Dialog



- Debug when workspace is open
Specifies the debug action just after the workspace opens.
- Save debug setup file (debug environment) when debug ends
Specifies whether to save setup information upon completion of debug session.
- Auto-load the targeted file after make/build (debug session)
Specifies whether to reload target file after make/build executed in debug session.

4.5.5 Setup Project

Setting regarding the project is performed.

■ Setup Project

The Setup Project dialog has the part where the target item is set and the part where setting is performed.

The set values are enabled when the [Apply] or [OK] button is clicked.

The directory can be described in a relative path from the project.

If two or more target items are set, the values are displayed as follows:

- Edit

If the values are equal in all target items, they are displayed as they are.

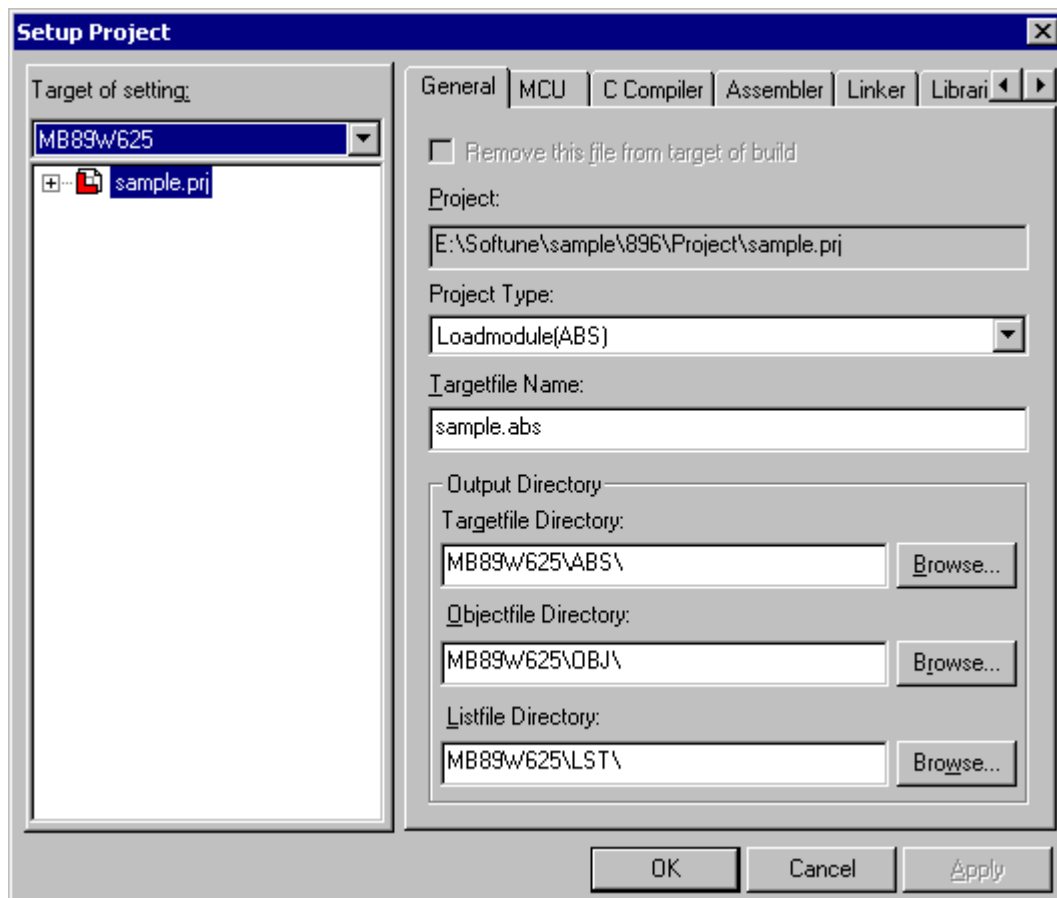
If the values are not equal in at least one target item, they are blanked.

- Check button

If the values are equal in all target items, they are displayed as they are.

If the values are not equal in at least one target item, they are dimmed.

Figure 4.5-6 Setup Project Dialog



■ Setup Target

The items to be set in the combo box and tree view at the left side of the dialog are specified.

- Target of setting-combo box:

All the configuration names of projects in workspace are displayed.

Configuration name: The selected configuration is set.

[All Configuration]: All configurations are set.

[Multiple Configuration]: The Multiple Configuration dialog (Figure 4.5-7) is opened. The two or more configurations specified in the dialog are set.

-Target of setting-Tree View:

All projects having the configuration names specified in the combo box are displayed. The items that can be set vary depending on the selected items. When two or more items are selected, they can be changed at a time. If the items of different types are selected, only overlapped items can be set.

Project: The [General] items, [MCU] items, common options ([C Compiler], [Assembler], [Linker], [Librarian]), and [Debug] items can be set.

C source file: The [General] items and individual options ([C Compiler]) can be set.

Assembler source file: The [General] items and individual options ([Assembler]) can be set.

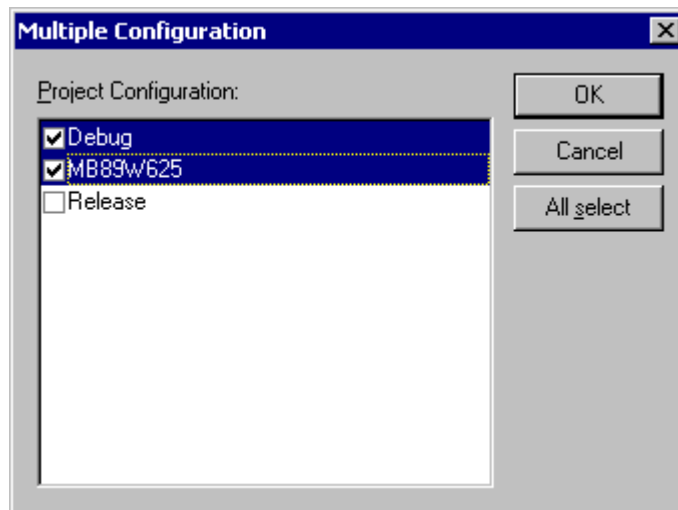
Library file: The [General] items can be set.

Object file: The [General] items can be set.

Relative file: The [General] items can be set.

Folder: All files in the folder are set.

Figure 4.5-7 Multiple Configuration Dialog



■ Setting Items

The following items can be set or changed in tab form.

- General

The project type, target name, output directory, and file building are set.

- MCU

The items regarding the MCU, such as the chip type and target MCU, are set.

- C Compiler

The C compiler options are set.

- Assembler

The assembler options are set.

- Linker

The linker options are set.

- Librarian

The librarian options are set.

- Converter

The converter start and converter options are set.

- Debug

The debug options and debugger setup information are set.

4.5.5.1 General

This section explains the items that can generally be set in the [Setup Project] dialog.

■ Setting of General

1. Click the [General] tab in the [Setup Project] dialog.
2. Set the following items as needed.

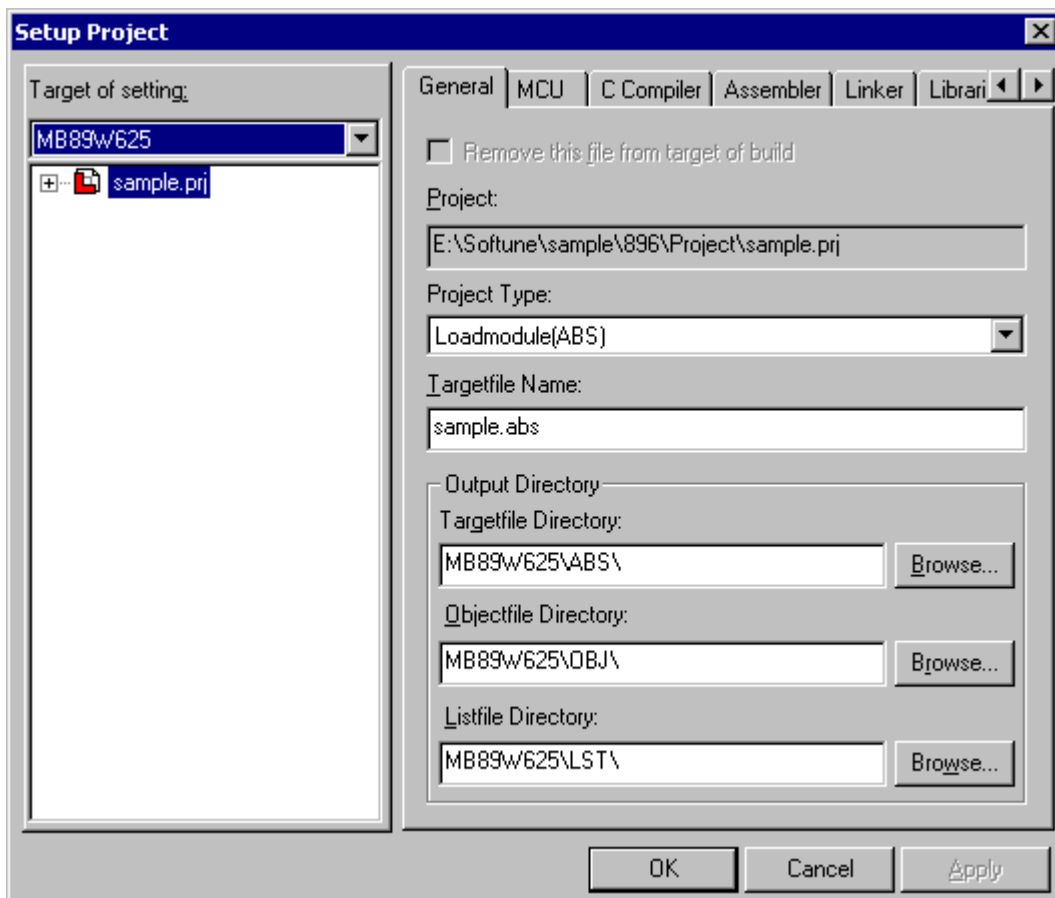
[Remove this file from target of Build]

[Project type]

[Output Directory]

Even though any category is selected, all the C compiler option currently being set is displayed at the bottom of the dialog.

Figure 4.5-8 Setup Project Dialog-General



- Remove this file from target of build

The file selected in the tree view is removed from the Make/Build target. This item can be set when C source file, Assembler source file, Library file, Object file, or Relative file is selected.

- Project

The full path to the project file is displayed.

- Project Type

Set the type of project file selected in the tree view (absolute (ABS)/relative (REL)/library (LIB)). If the REALOS (ABS) type is selected, the project type cannot be changed. This item can be set when the project file is selected singly.

The project type cannot be set for each configuration.

- Target File Name

Set the main file name of the file. This item can be set when only Project (two or more items may be set) is selected.

- Target File Directory

Set the directory of the target file. This item can be set when only Project (two or more items may be set) is selected.

- Object File Directory

Set the output directory of the object file output in compiling or assembling. This item can be set when only Project (two or more items may be set) is selected.

- List File Directory

Set the output directory of the list file output in compiling, assembling, or building. This item can be set when only Project (two or more items may be set) is selected.

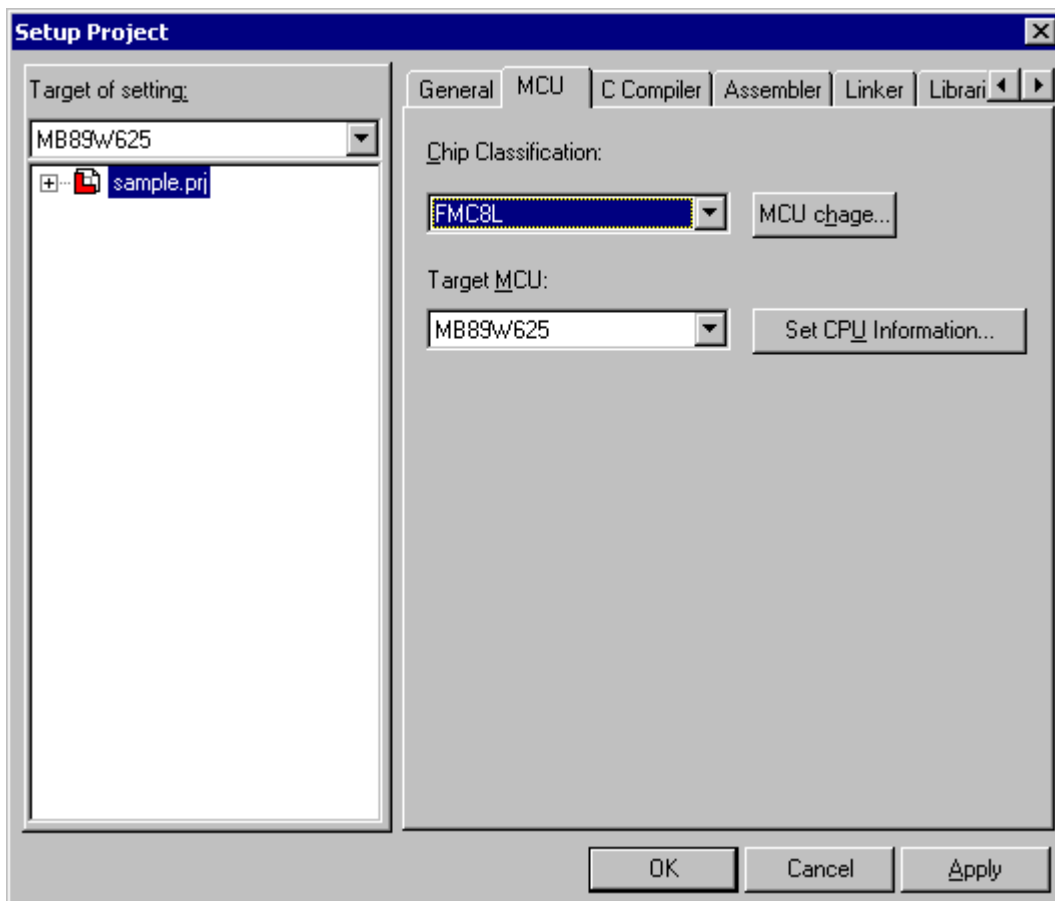
4.5.5.2 MCU

This section explains the items that can be set MCU in the [Setup Project] dialog.

■ Setting of MCU

1. Click the [MCU] tab in the [Setup Project] dialog.
2. Set the following items as needed: [Chip Classification], [Target MCU], [Set CPU information], [MCU Change].

Figure 4.5-9 Setup Project Dialog-MCU

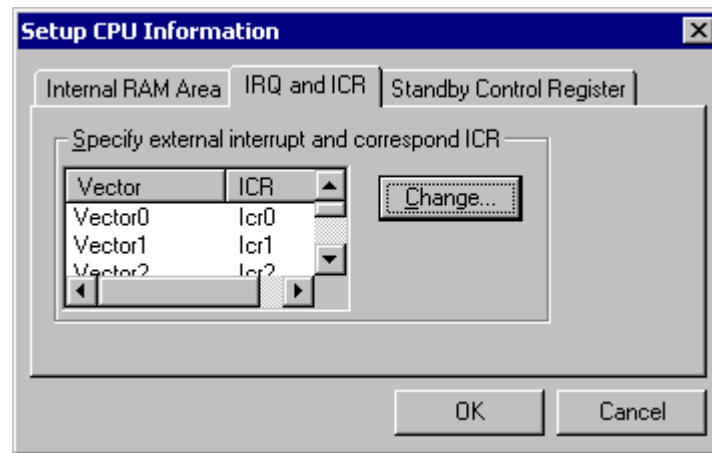


- Chip Classification
Selects the type of chip.
- Target MCU
Select the target MCU of the chip type selected in [Chip Classification].
- Set CPU Information
The Set CPU Information dialog is opened.
- MCU Change
Open MCU Change dialog.

■ Setting of CPU Information

- Setup CPU information - IRQ and ICR

Figure 4.5-10 CPU Information Setup Dialog (IPQ and ICR)



- Relationship between Vector and ICR

The correspondence between external interrupts (Vector) and the interrupt control register (ICR) varies with the target MCU.

This correspondence is specified by the simulator debugger.

- Setting Correspondence between External Interrupts and ICRs

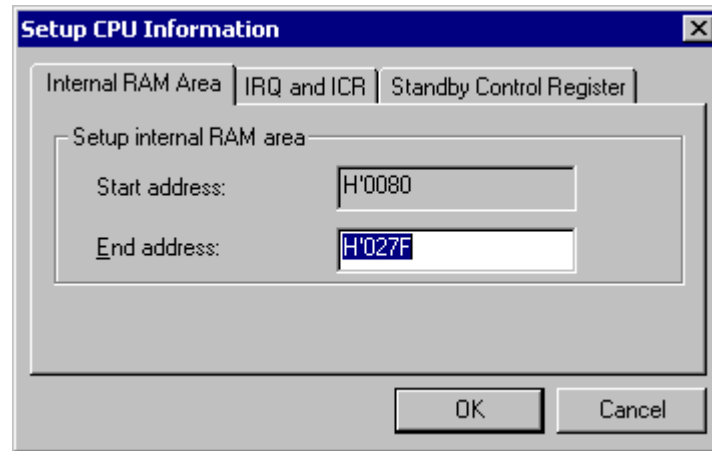
Set the interrupt control registers (ICRs) corresponding to vector numbers. When a vector number is selected and the [Change] button is clicked, the dialog shown in Figure 4.5-11 opens.

Figure 4.5-11 Correspondence between External Interrupts and ICRs



- Setup CPU Information - Internal RAM Area

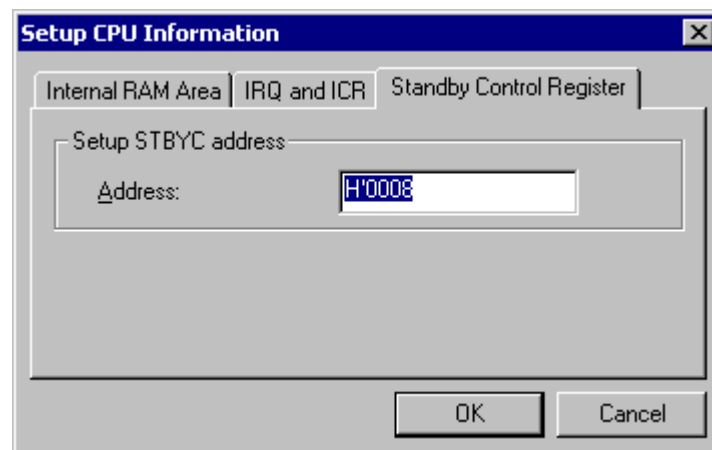
Figure 4.5-12 CPU Information Setup Dialog (Internal RAM Area)



- Start Address
Sets start address of internal RAM area.
- End Address
Sets end address of internal RAM area.

- Setup CPU Information - Standby Control Register

Figure 4.5-13 CPU Information Setup Dialog (Standby Control Register)



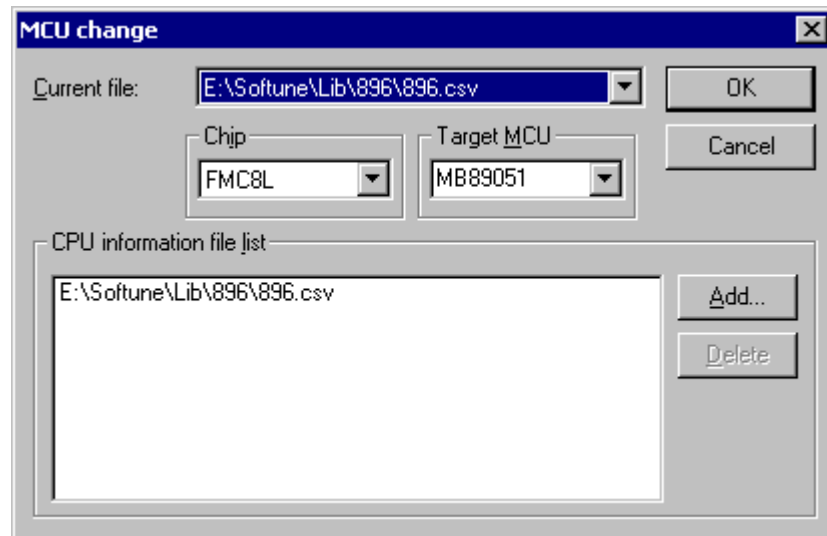
- Address
Sets STBYC address.

■ Setting of MCU Change Dialog

1. Click [MCU change] button.
The MCU change dialog shown in Figure 4.5-10 opens.
2. From [Current file], choose the CPU information file to be used. If a CPU information file to be used is not found, add it with the [Add] button for [CPU information file list].
The MCU list for the selected CPU information file appears in [Chip] and [Target MCU].

3. Choose the MCU to be changed from the [Chip] and [Target MCU], and click [OK] button.
The change you specified takes effect.

Figure 4.5-14 MCU Change Dialog



- Current file
Choose the CPU information file to be used. The MCU list for the selected CPU information file appears in [Chip] and [Target MCU].
- Chip
Choose a chip type.
- Target MCU
Choose a target MCU for the chip type that is chosen from the [Chip] area.
- CPU information file list
Lists the registered CPU information files.
- Add
Adds a CPU information file.
- Delete
Deletes a CPU information file that is chosen from the [CPU information file list].
Note that the files that are retrieved by default cannot be deleted.

Note:

There are no functions with some CPU.

4.5.5.3 Setting C Compiler Options

This section explains how to set C compiler options.

■ Setting C Compiler Options

1. Click the [C Compiler] tab from the project setup dialog.
2. Select category.

[General], [Define Macro], [Include Path], [Optimize] or [Language specifications] can be selected as category.

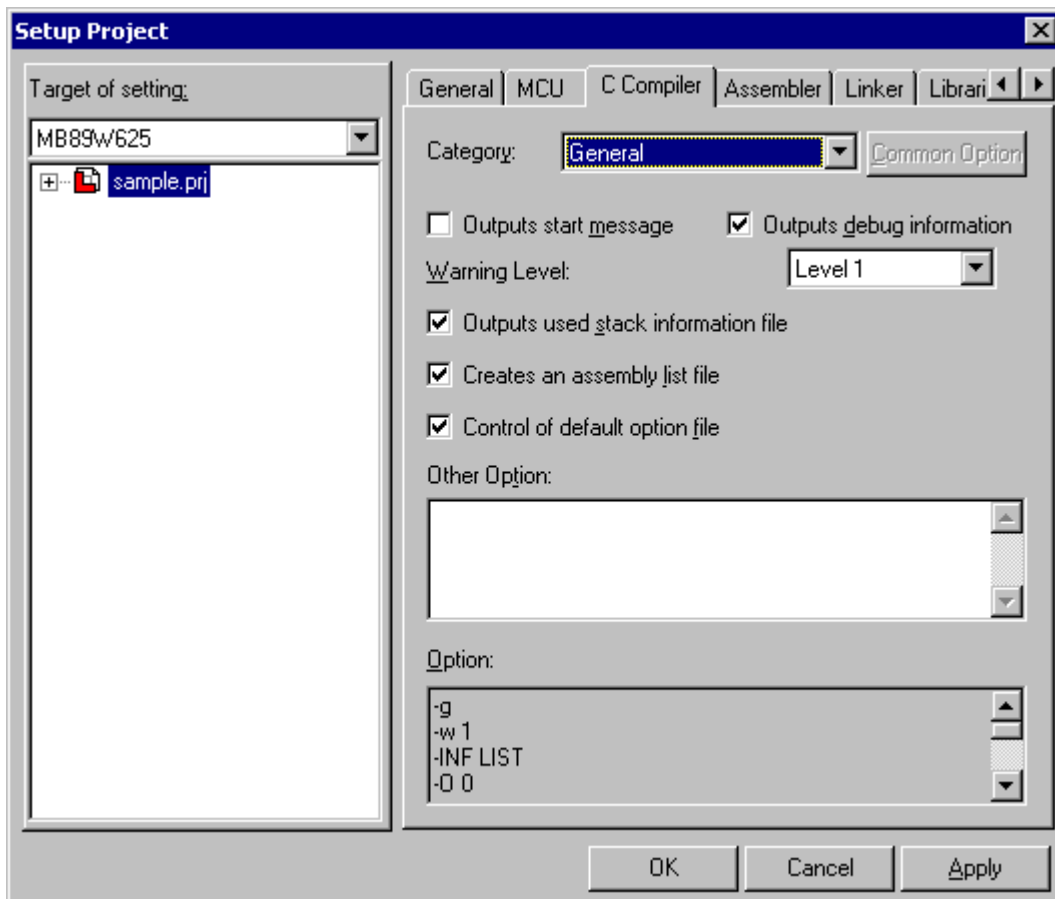
Even though any category is selected, all the C compiler option currently being set is displayed at the bottom of the dialog.

If the individual option is set, the [Common Option] button can be clicked to return to the common option.

The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE Workbench User's Manual.

■ Setting [General] Options

Figure 4.5-15 General Option Setup Dialog



The following options can be set from the normal option setup dialog.

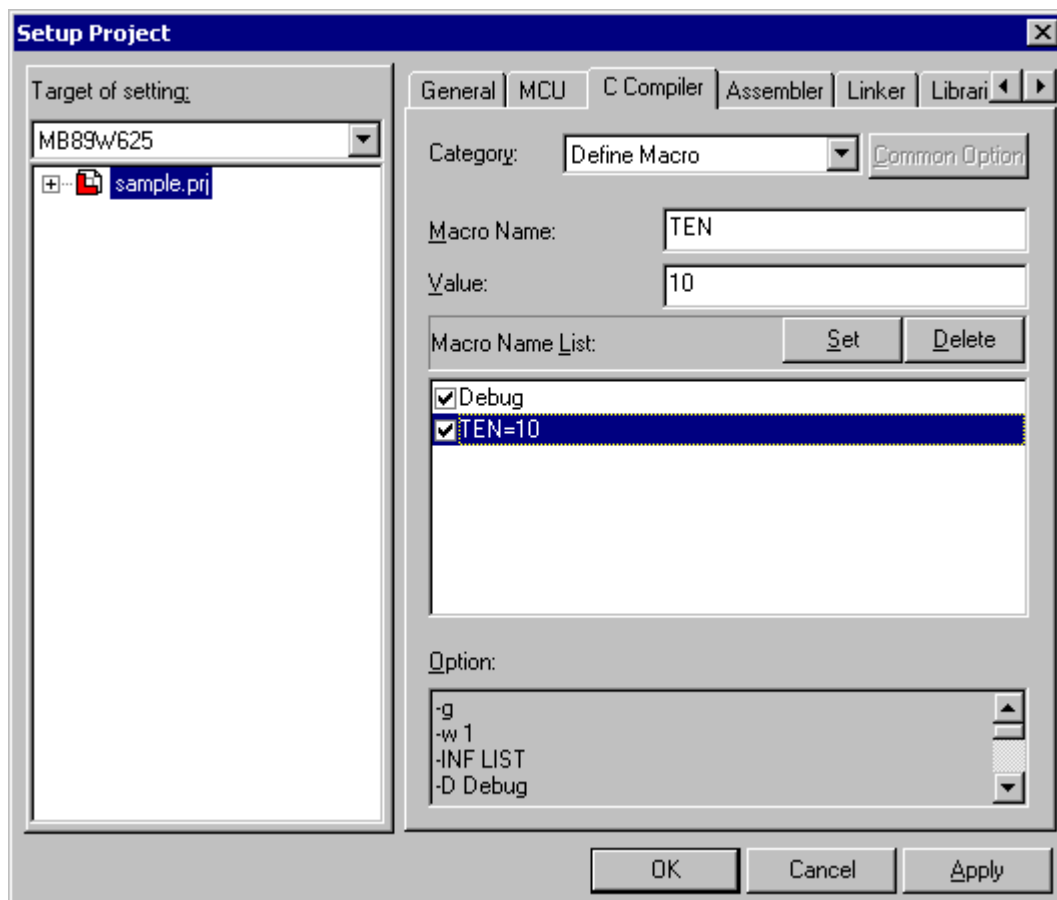
- Outputs start message (-V).
- Outputs debug information (-g).
- Outputs warning message (or warning level) (-w).
- Outputs stack usage information file (-INF STACK).
- Creates assembly list file (-INF LIST).
- Control of default option file (-Xdof).

Depending on CPU classification, [Output warning level] (levels 0 to 3) may be selected instead of [Output warning message].

At Specification in [Other Option], all C compiler options can be written like start-up options from command lines. Write the options that do not belong to any C compiler option setup categories directly at Specification in [Other Option].

■ Setting a Macro Name

Figure 4.5-16 Macro Name Setup Dialog



If there are two or more items to be set, the macro name found in some item is grayed.

1. Select the [Define Macro] category.
The macro name setup dialog shown in Figure 4.5-16 opens.
2. Specify the [Macro Name].
3. Specify the [Value] as required.
4. Click the [Set] button.

The specified macro name is set as a define (-D) option.

5. To set the specified macro name as an undefine (-U) option, reset the check mark of the macro name from [Macro Name List].

Note:

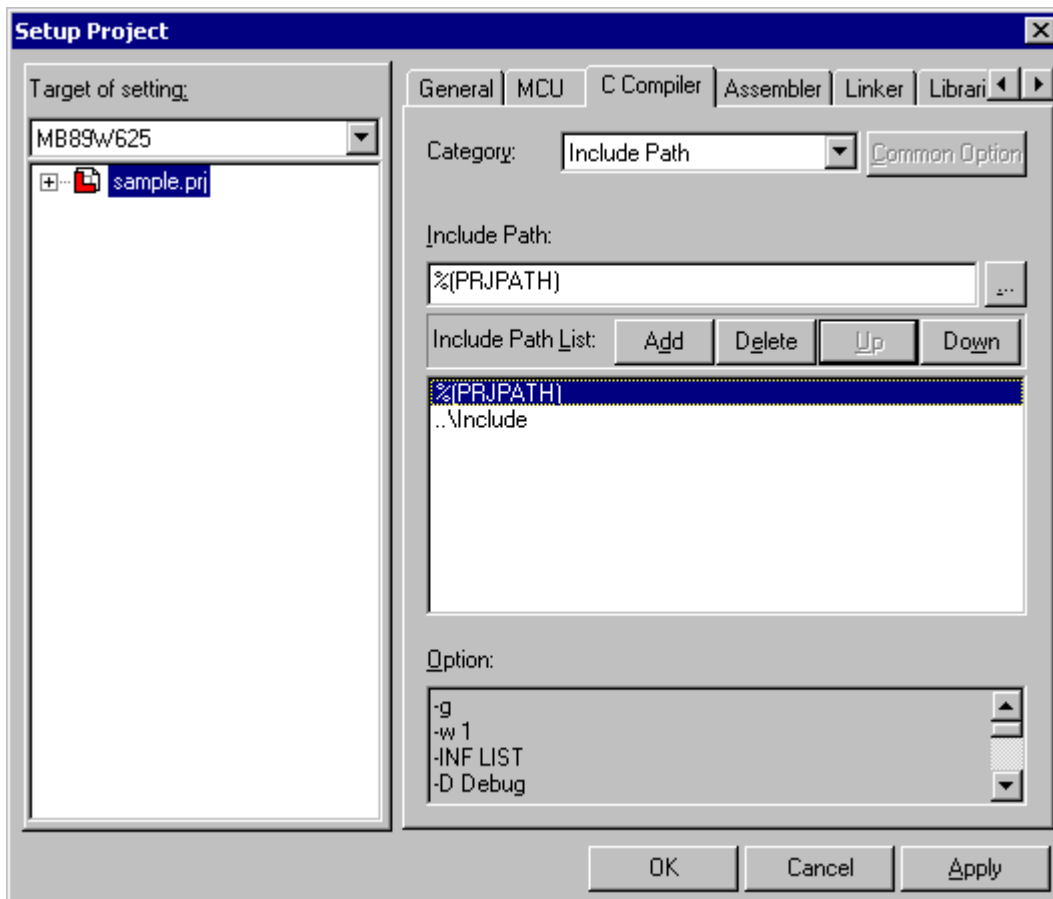
When "undefine" is set, both the define and undefine options are output for the same macro name. This causes no problem because the undefine option precedes the define option.

■ Resetting a Macro Name

1. Select the [Define Macro] category.
The macro name setup dialog shown in Figure 4.5-16.
2. Select the macro name to reset from [Macro Name List].
3. Click the [Delete] button.

■ Setting an Include Path

Figure 4.5-17 Include Path Setup Dialog



1. Select the [Include Path] category.
The include path setup dialog shown in Figure 4.5-17 opens.
2. Specify the include path.
Clicking the [Brows...] button to the right of the input field enables directory selection.

3. Click the [Add] button.

The specified include path is added to the end of the [Include Path List].

■ Resetting an Include Path

1. Select the [Include Path] category.

The include path setup dialog shown in Figure 4.5-17 opens.

2. Select the include path to reset in the [Include Path List].
3. Click the [Delete] button.

■ Changing the Include Path Retrieval Order

Include paths are retrieved in turn from top of the [Include Path List]. The order in which include paths are registered in the list can be changed as follows:

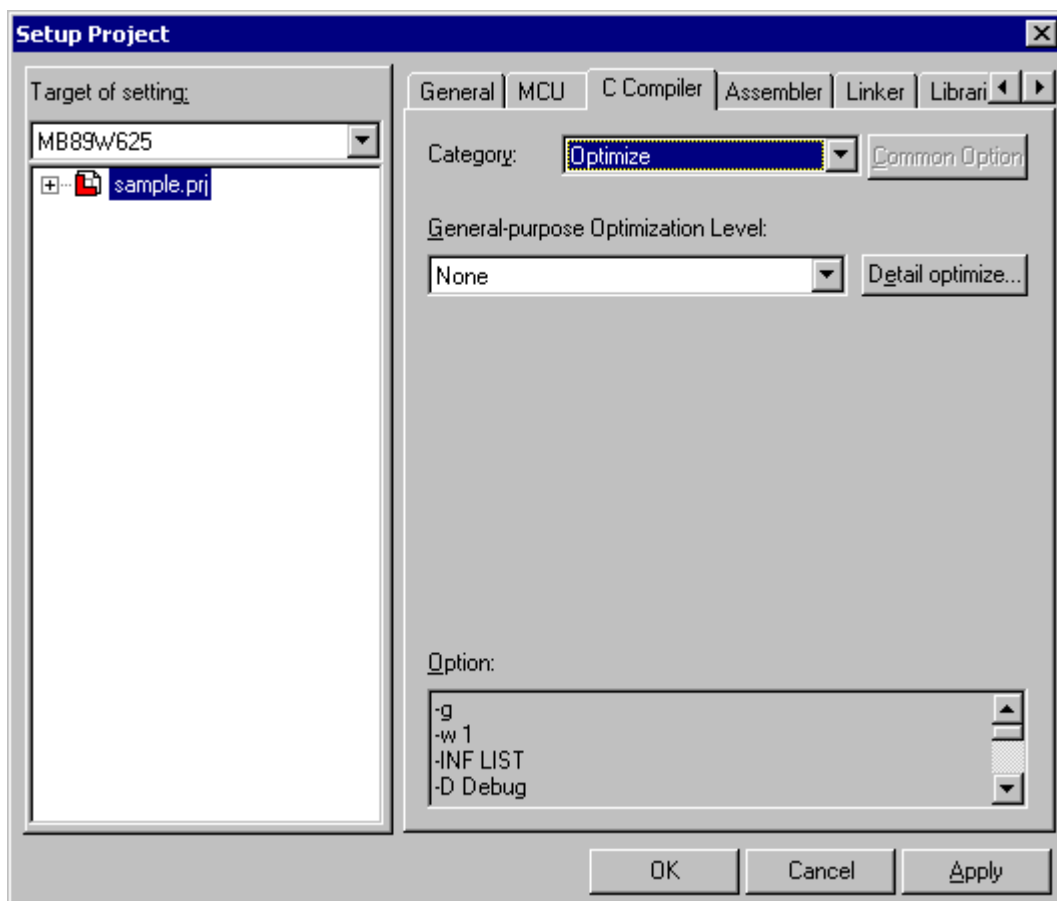
1. Select the [Include Path] category.

The include path setup dialog shown in Figure 4.5-17 opens.

2. Select the include path whose order is to be changed from [Include Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting [Optimize] Options

Figure 4.5-18 Optimize Option Setup Dialog

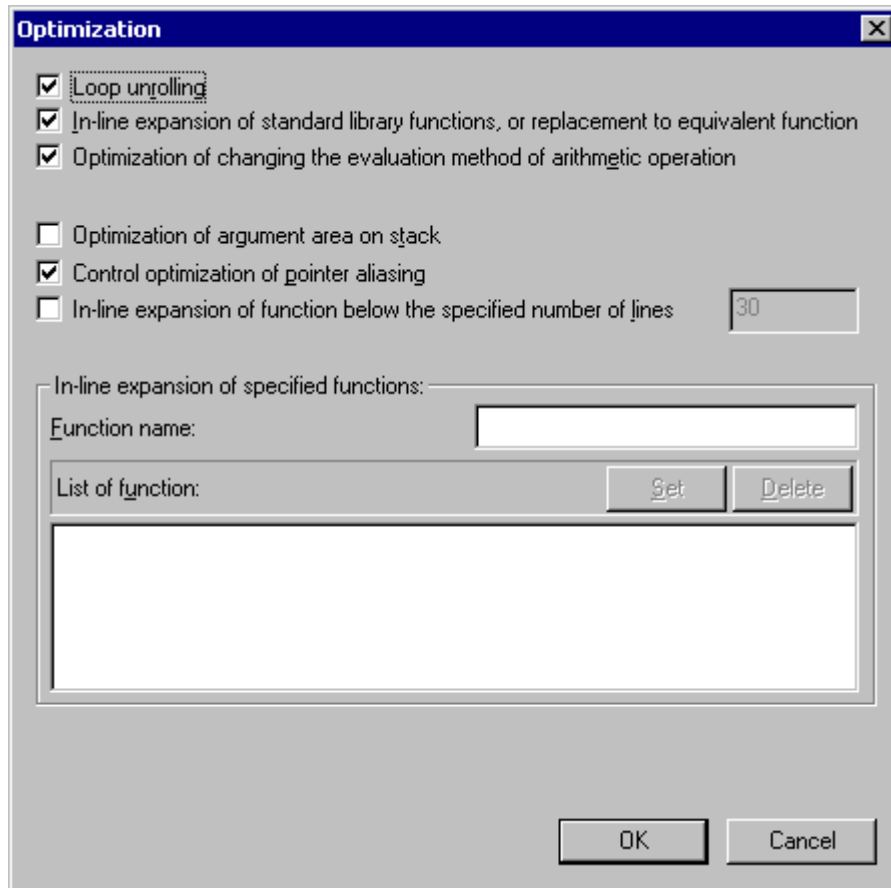


The following options can be set from the optimize option setup dialog. Select the optimize option to set from the drop-down list.

- General-purpose optimization level. (-O)
(None/Level1-4/speed priority/size priority)

■ Setting of Options Included in [Detail Definition] in [Optimization]

Figure 4.5-19 Optimize Details Setup Dialog



The following options can be specified:

- Loop unrolling (-K UNROLL)
- In-line expansion of standard library functions, or replacement to equivalent function (-K LIB)
- Optimization of changing the evaluation method of arithmetic operations (-K EOPT)
- Optimization of argument area on stack (-K ADDSP)
- Control optimization of pointer aliasing (-K ALIAS)
- In-line expansion of function below the specified number of lines (-x auto)
- In-line expansion of specified functions (-x)

■ Setting In-line Expansion of Specified Functions

1. Select the [Optimization] category.
The Set Optimization dialog (See Figure 4.5-19) opened.
2. Set [General-purpose Optimization Level] to any values other than 0 and click [Detail optimize] button.
3. Specify [Function name].
4. Click the [Set] button.

■ Canceling In-line Expansion of Specified Functions

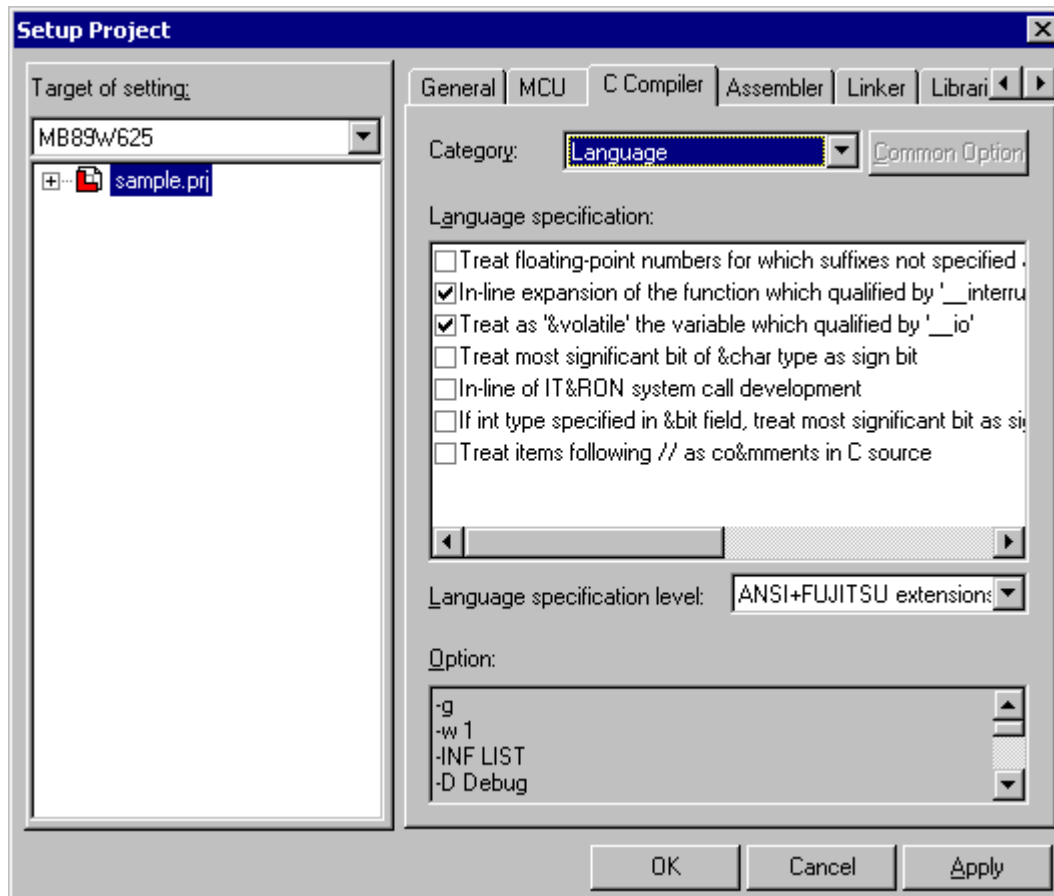
1. Select the [Optimization] category.
The Set Optimization dialog (See Figure 4.5-19) opened.
2. Sets [General-purpose Optimization Level] to any values other than 0 and click [Detail optimize] button.
3. Select the function name to be deleted from [List of function].
4. Click the [Delete] button.

Note:

If an optimization level is changed, options set in the Detail Definition Dialog are initialized according to the optimization level.

■ Setting of Options Included in [Language Specifications]

Figure 4.5-20 Language Specification Option Setup Dialog



The following options can be specified:

- Treat floating-point numbers for which suffixes are not specified as float type (-K FCONST/DCONST).
- In-line expansion of the function which is qualified by '_interrupt' (-K NOINTLIB).
- Treat the variable which is qualified by '_io' as 'volatile' (-K NOVOLATILE).
- Treat most significant bit of char type as sign bit (-K SCHAR).

- In-line of ITRON system call development (-K REALOS).
- If int type specified in bit field, treat most significant bit as sign bit (-K SBIT).
- Change order in which & static variables are stored in memory to order in which sources are described (-verorder).
- Treat items following " // " as comments in C source (-B).
- Language specification level (-Ja|c|e)
(ANSI/ANSI + FUJITSU extensions)

4.5.5.4 Setting Assembler Options

This section explains how to set assembler options.

■ Setting Assembler Options

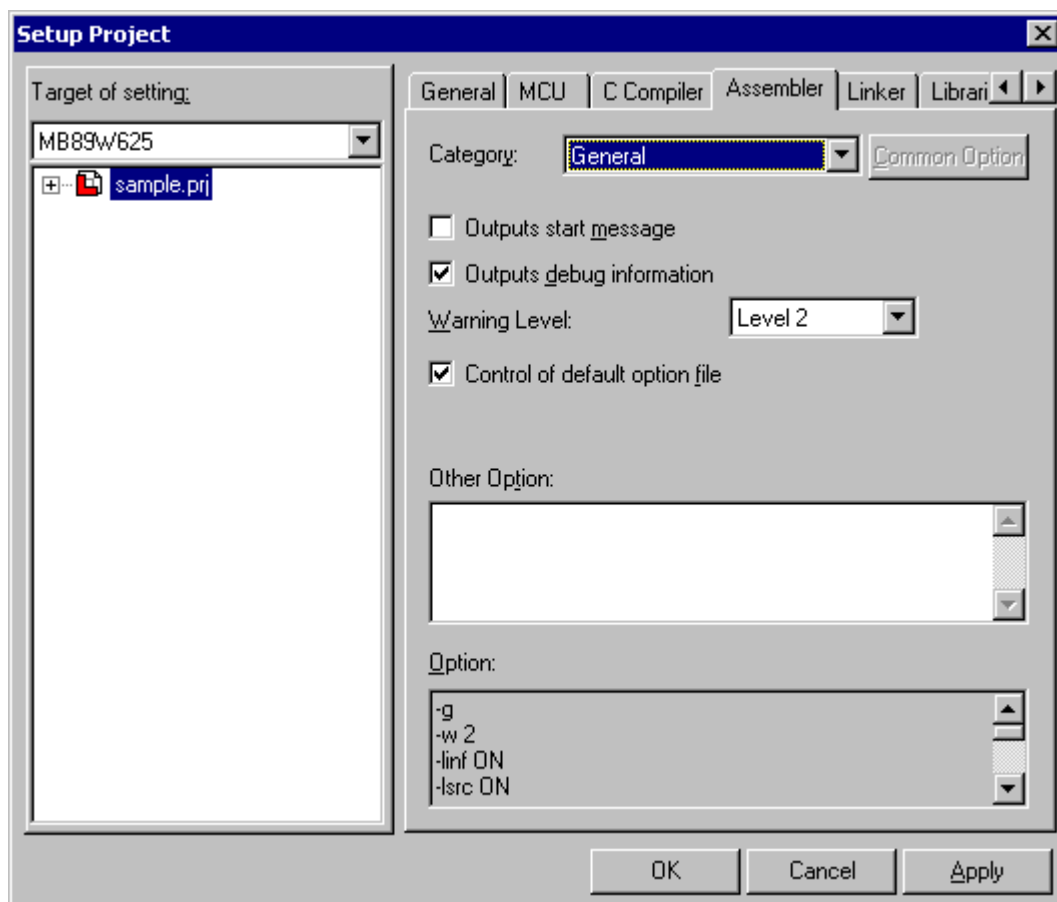
1. Click the [Assembler] tab from the project setup dialog.
2. Select category.

[General], [Define Macro], [Include Path], [Target Depend], or [Output List] can be selected as category. Specified options can be checked using [Option] at the bottom of the dialog.

If the individual option is set, the [Common Option] button can be clicked to return to the common option. The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Description Usable in Manager" of SOFTUNE Workbench User's Manual.

■ Setting [General] Options

Figure 4.5-21 General Option Setup Dialog



The following options can be set from the general option setup dialog.

- Outputs start message (-V)
- Outputs debug information (-g)

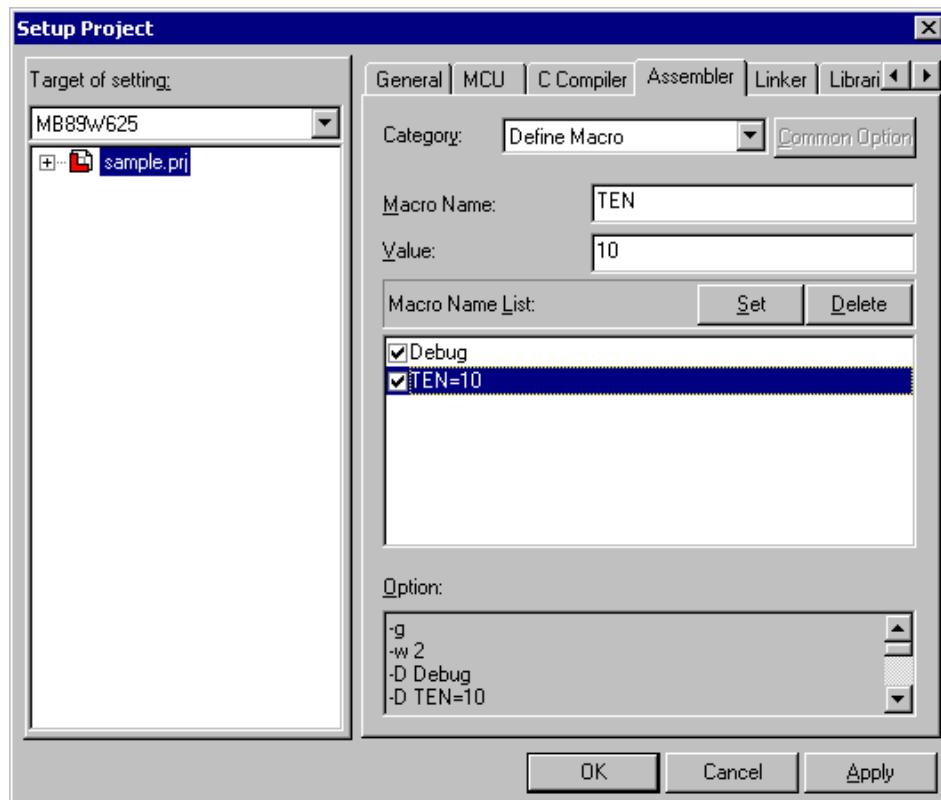
- Control of default option file (-Xdof)
- Warning Level (-w) (Level 0 to 3)

Select [Warning Level] from the drop-down list.

In [Other Option], all assembler options can be written like startup options from command lines. Write the options that do not belong to any assembler option setup categories directly in [Other Option].

■ Setting a Macro Name

Figure 4.5-22 Macro Name Setup Dialog



If there are two or more items to be set, the macro name found in some item is grayed.

1. Select the [Define Macro] category.
The macro name setup dialog shown in Figure 4.5-22 opens.
2. Specify a [Macro Name].
3. Specify a [Value] as required.
4. Click the [Set] button.
The specified macro name is set as a define (-D) option.
5. To set the specified macro name as an undefine (-U) option, reset the check mark of the macro name in [Macro Name List].

Note:

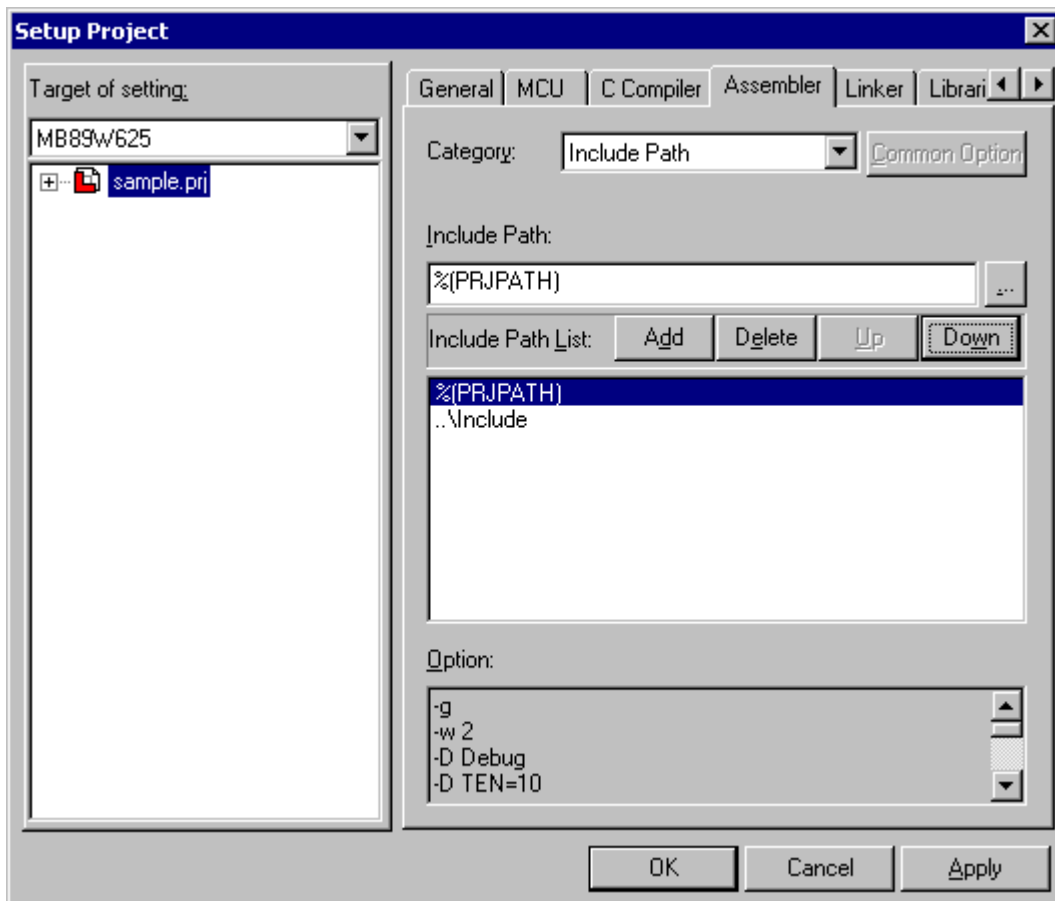
When "undefine" is set, both the define and undefine options are output for the same macro name. This causes no problem because the undefine option precedes the define option.

■ Resetting a Macro Name

1. Select the [Define Macro] category.
The macro name setup dialog shown in Figure 4.5-22 opens.
2. Select the macro name you want to reset from [Macro Name List]
3. Click the [Delete] button.

■ Setting an Include Path

Figure 4.5-23 Include Path Setup Dialog



1. Select the [Include Path] category.
The include path setup dialog shown in Figure 4.5-23 opens.
2. Specify an include path.
Clicking the [Browse] button to the right of the input field enables directory selection.
3. Click the [Add] button.
The specified include path is added to the end of [Include Path List].

■ Resetting an Include Path

1. Select the [Include Path] category.
The include path setup dialog shown in Figure 4.5-23 opens.
2. Select the include path you want to reset from [Include Path List]
3. Click the [Delete] button.

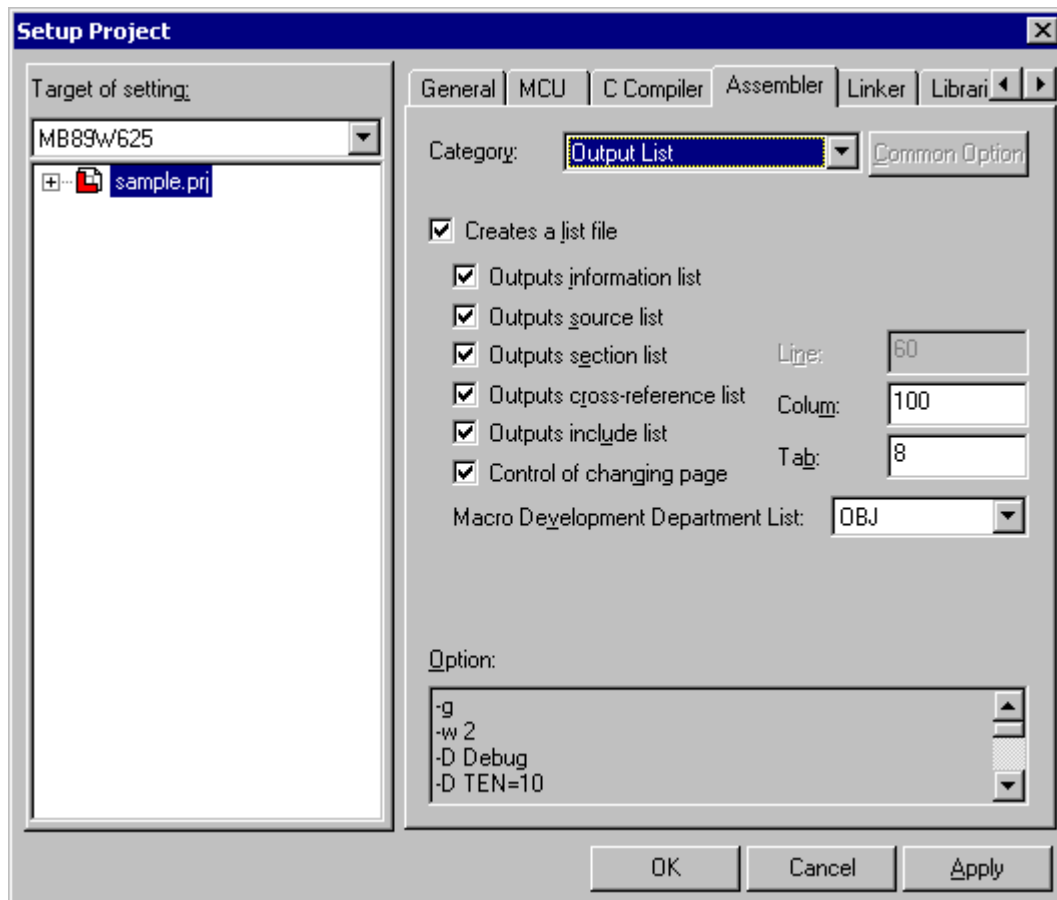
■ Changing the Include Path Retrieval Order

Include paths are retrieved in turn from top of [Include Path List]. The order in which include paths are registered in the list can be changed in the following procedure:

1. Select the [Include Path] category.
The include path setup dialog shown in Figure 4.5-23 opens.
2. Select the include path whose order is to be changed from [Include Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting List Output

Figure 4.5-24 List Output Setup Dialog

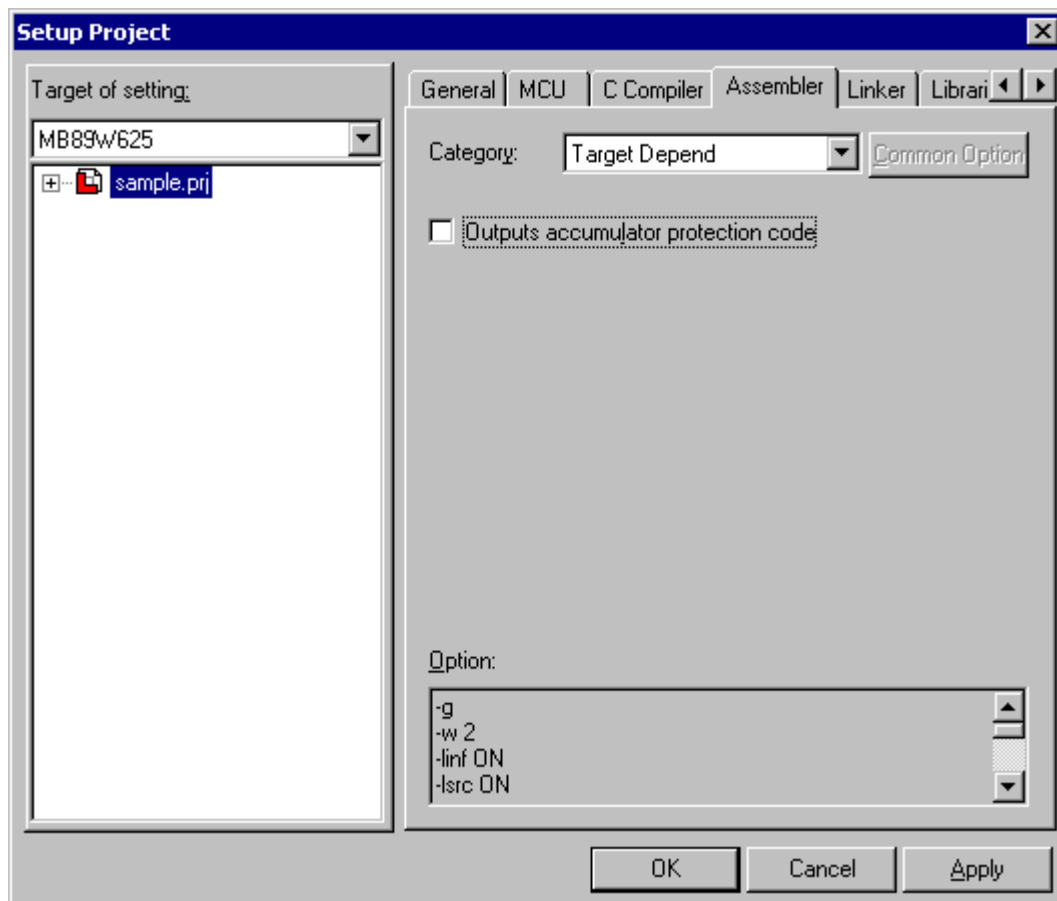


1. Select the [Output List] category.
The list output setup dialog shown in Figure 4.5-24 opens.
2. To output a list file, set a check mark to the left of [Creates a list file].
When the list file is not output, no other item need to be set. Execute Step (3) and after only when outputting the list file.
3. Select the list file(s) you want to output. Items with check marks are selected.
 - Outputs information list
 - Outputs source list
 - Outputs section list
 - Outputs cross-reference list
 - Outputs include list

4. Specify whether to suppress page change. When a check mark is set to the left of [Control of changing page], page change is suppressed.
When page change is suppressed, the line count cannot be set.
5. Select a [Macro Development Department List] from the drop-down list. (No output, SRC/OBJ, OBJ)
6. Set line count, column count, and tab count to the right of [Line], [Column], and [Tab] as required.

■ [Target Depend] Options

Figure 4.5-25 Target Dependency Setup Dialog



The following options can be specified:

- Outputs accumulator protection code (-sa)

4.5.5.5 Setting Linker Options

This section explains how to set linker options.

■ Setting Linker Options

1. Click the [Linker] tab from the project setup dialog.
2. Select category.

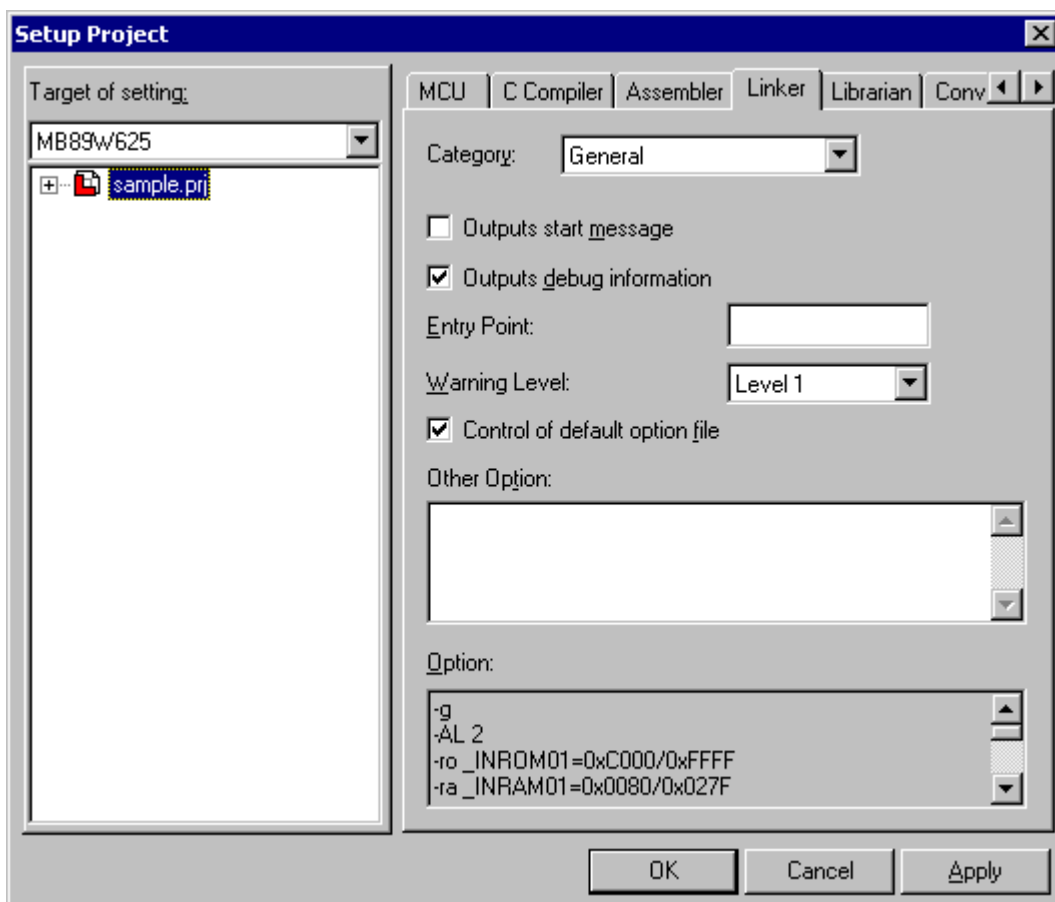
[General], [Allocation/Link], [Define Symbol], [List Output], [Absolute Assemble List], [Control Library], or [Register Bank] can be selected as category.

Specified options can be checked using [Option] at the bottom of the dialog.

The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE Workbench User's Manual.

■ Setting [General] Options

Figure 4.5-26 General Option Setup Dialog



The following options can be set from the general option setup dialog.

- Outputs start message (-V)
- Outputs debug information (-g)

- Control of Default Option File (-Xdof)
- Entry Point (-e)
- Warning Level (-w)(Level 0 to 2)
- Other Option

[Entry Point] is the address to be set in the PC when data is loaded by the debugger. Be sure to specify this option with a global symbol.

Select [Warning Level] from the drop-down list.

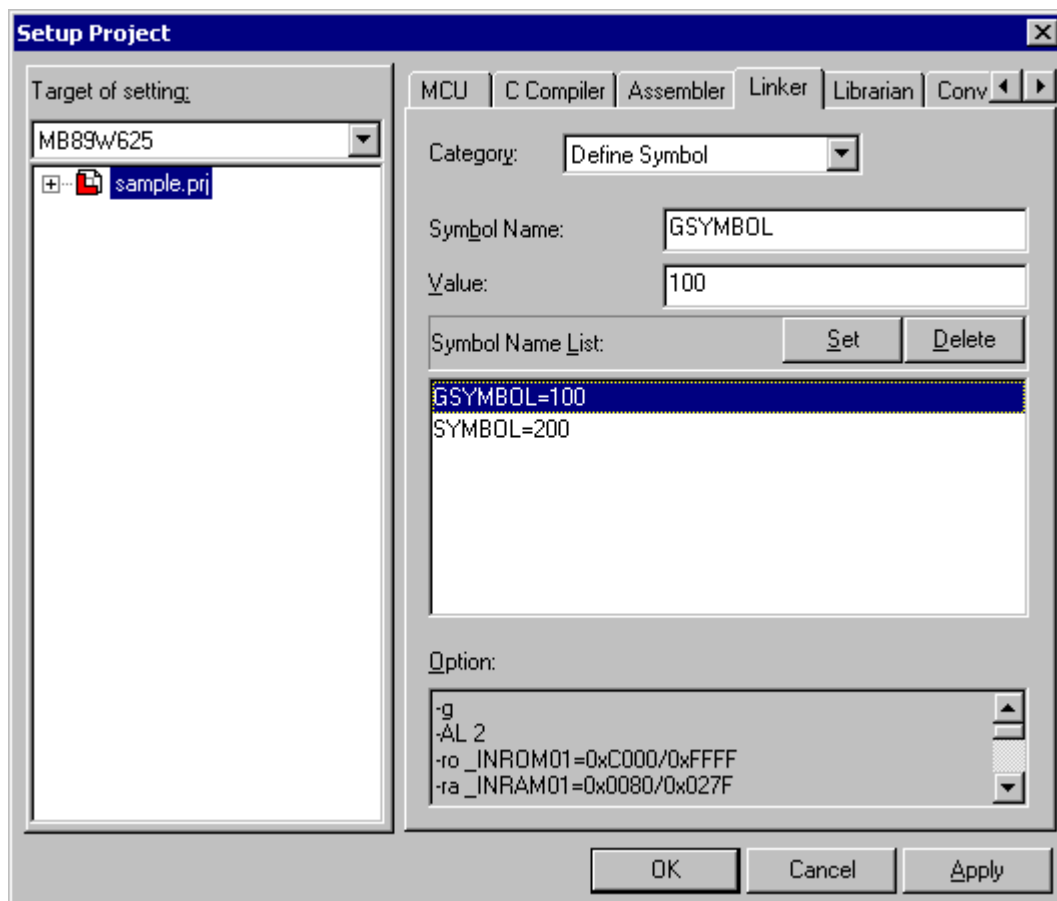
In [Other Option], all linker options can be written like startup options from command lines. Write the options that do not belong to any linker option setup categories directly in [Other Option].

■ Setting Disposition/Connection

See Section "4.5.5.6 Section Disposition/Connection Specifying".

■ Setting Symbol Definition

Figure 4.5-27 Symbol Definition Setup Dialog



If there are two or more items to be set, the symbol name found in some item is grayed.

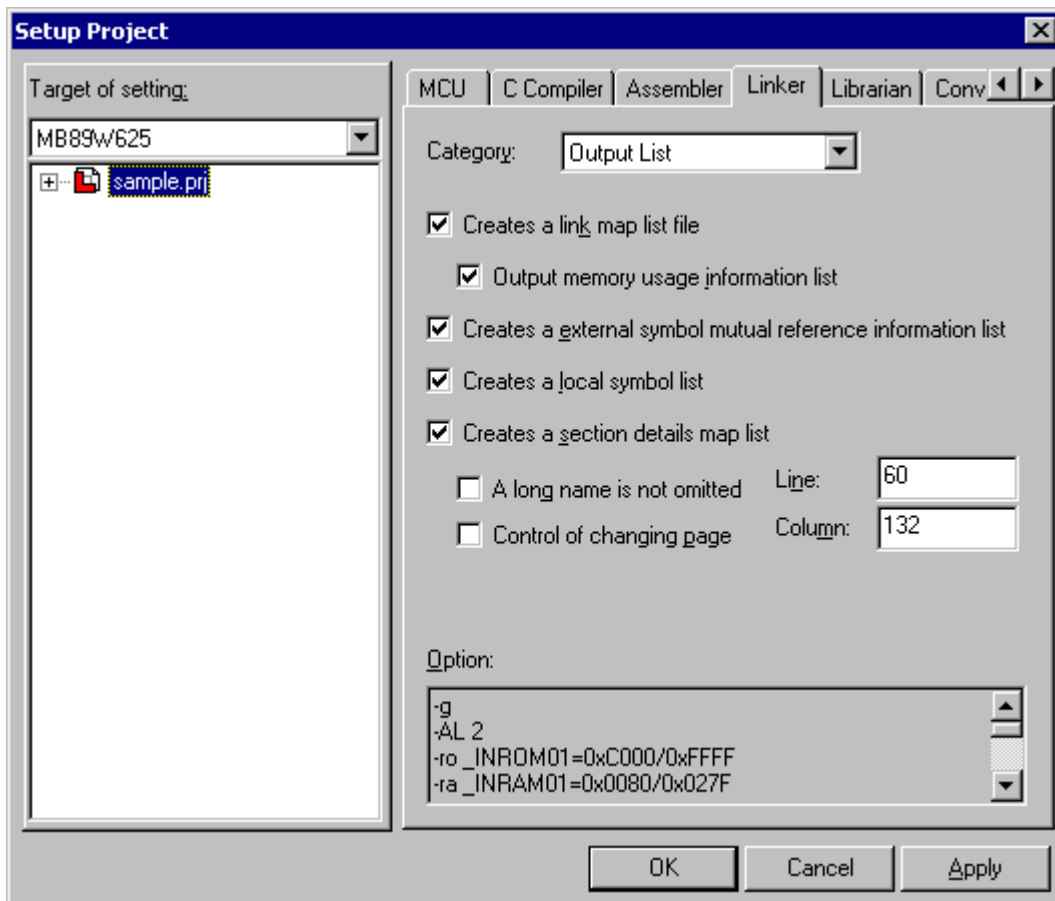
1. Select the [Define Symbol] category.
The symbol definition setup dialog shown in Figure 4.5-27 opens.
2. Specify a [Symbol Name].
3. Specify a [Value].
4. Click the [Set] button.

■ Resetting Symbol Definition

1. Select the [Define Symbol] category.
The symbol definition setup dialog shown in Figure 4.5-27 opens.
2. Select the symbol name you want to reset from [Symbol Name List].
3. Click the [Delete] button.

■ Selecting Output File

Figure 4.5-28 Output List Setup Dialog



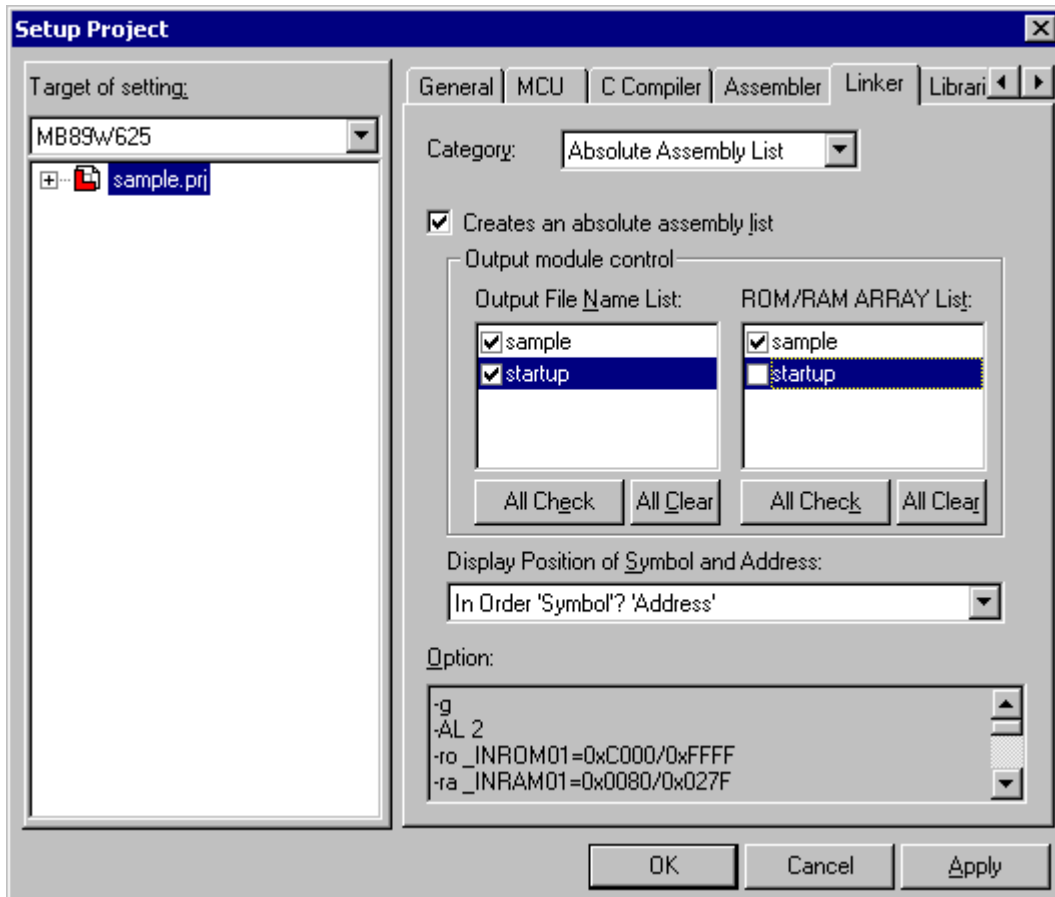
1. Select the [Output List] category.
The list output setup dialog shown in Figure 4.5-28 opens.
2. Set check marks to the left of the list(s) to be created.
 - Creates a link map list file
 - Creates a external symbol mutual reference information list
 - Creates a local symbol list
 - Creates a section details map list

When none of the above lists is output, no other item need to be set from the list output setup dialog. Execute Step (3) and after only when outputting any of the lists.
3. Check [Output memory usage information list] as needed.
4. Set a check mark to the left of [A long name is not omitted] as required.
Even names exceeding one list file line are fully output.

5. Specify whether to suppress page change. When a check mark is set to the left of [Control of changing page], page change is suppressed.
When page change is suppressed, the line count cannot be set.
6. Set line count and column count to the right of [Line] and [Column] as required.

■ Setting Absolute Format Assemble List Options

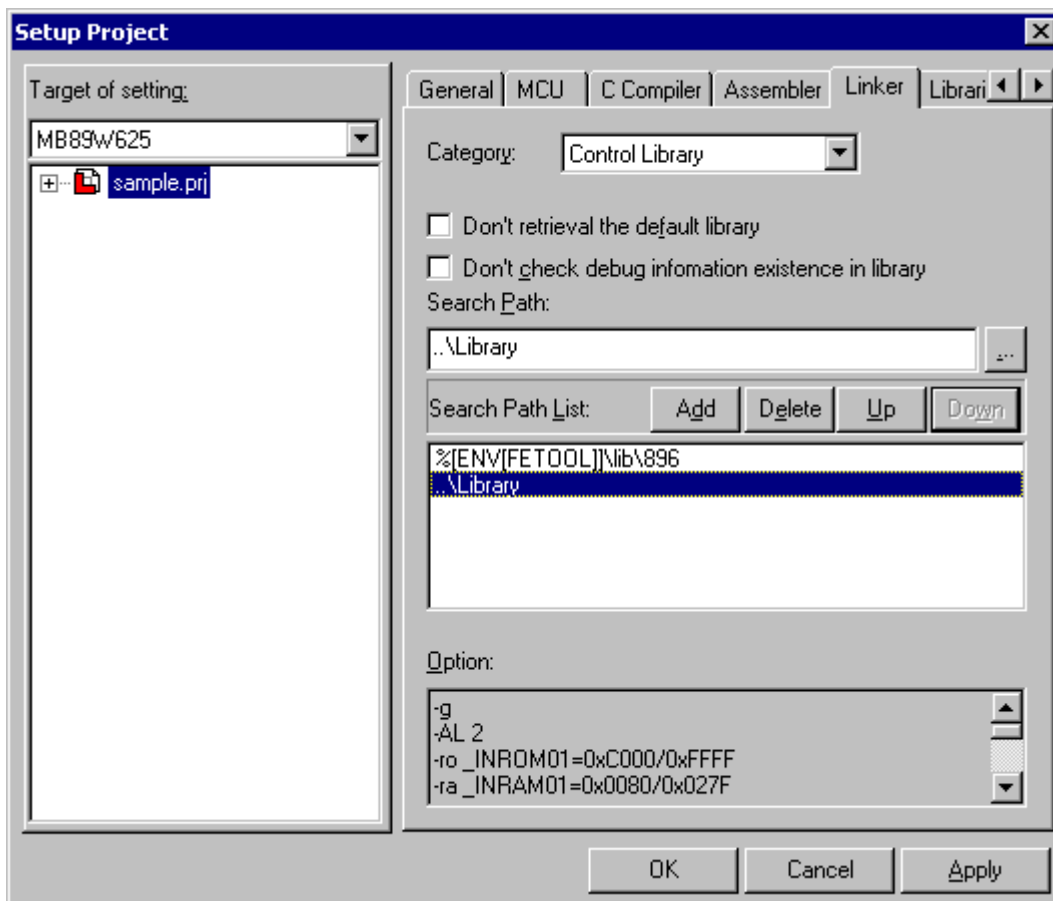
Figure 4.5-29 Absolute Format Assemble List Setup Dialog



1. Select the [Absolute Assembly List] category.
The absolute format assemble list setup dialog shown in Figure 4.5-29 opens.
2. Set a check mark to the left of [Creates an absolute assembly list].
When the absolute format assemble list file is not output, no other item need to be set from the absolute format assemble list setup dialog. Execute Step (3) and after only when outputting the absolute format assemble list file.
3. Select the module to which the absolute format assemble list is to be output from [Output File Name List], then set a check mark to the left of the selected module.
4. Select the module to which the ROM/RAM ARRAY list is to be output from [ROM/RAM ARRAY List], then set a check mark to the left of the selected module.
5. Select [Display Position of Symbol and Address] from the drop-down list.
(In Order 'Address' to 'Symbol' / In Order 'Symbol' to 'Address')

■ Setting of Options for Library Control

Figure 4.5-30 Library Control Setup Dialog



■ Setting When the Default Library is not Used

1. Select the [Control Library] category.
The library control setup dialog shown in Figure 4.5-30 opens.
2. Set a check mark to the left of [Don't retrieve the default library].

■ Setting Method When Debug Information Existence Check for Library File Module is Inhibited

1. Select the [Control Library] category.
The library control setup dialog shown in Figure 4.5-30 opens.
2. Set a check mark to the left of [Don't check debug information existence in library].

■ Setting a Library Retrieval Path

1. Select the [Control Library] category.
The library control setup dialog shown in Figure 4.5-30 opens.
2. Set [Search Path].
Clicking the [Browse] button to the right of the input file enables path reference.
3. Click the [Add] button.
The set library retrieval path is added to the end of [Search Path List].

■ Resetting a Library Retrieval Path

1. Select the [Control Library] category.
The library control setup dialog shown in Figure 4.5-30 opens.
2. Select the path you want to reset from [Search Path List].
3. Click the [Delete] button.

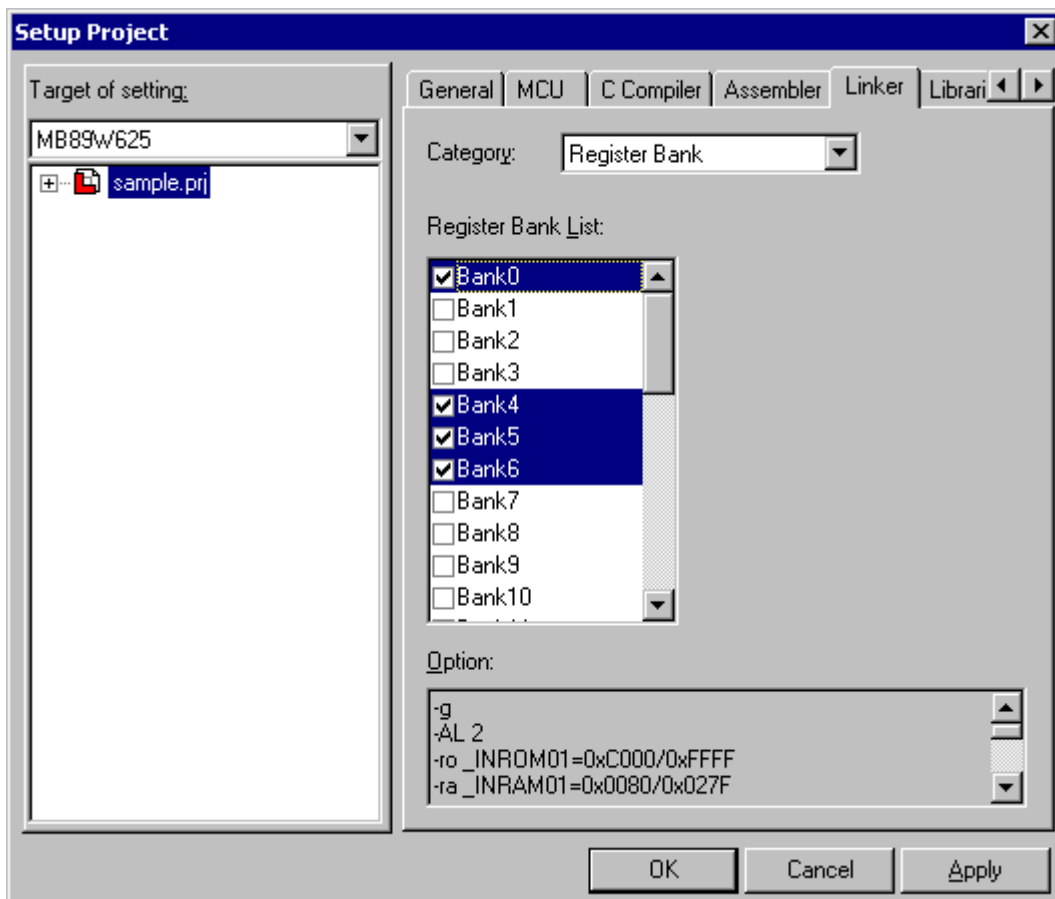
■ Changing the Library Retrieval Path Retrieval Order

Library retrieval paths are retrieved in turn from top of [Search Path List]. The order in which library retrieval paths are registered in the list can be changed in the following procedure:

1. Select the [Control Library] category.
The library control setup dialog shown in Figure 4.5-30 opens.
2. Select the library retrieval path whose order is to be changed from [Search Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting Register Bank

Figure 4.5-31 Register Bank Setup Dialog



1. Select the [Register Bank] category.
The register bank setting dialog is displayed (Figure 4.5-31).
2. Set the check mark for the register bank list to be used.

4.5.5.6 Section Disposition/Connection Specifying

This section explains how to set the section allocation/link option (linker option).

■ Section Disposition/Connection Specifying

1. Click the [Linker] tab from the project setup dialog.
2. Select the [Disposition/Connection] category.
The disposition/connection setup dialog shown in Figure 4.5-32 opens.
3. Select [Auto Disposition] from the drop-down list (None/Mode1/Mode2).

- NONE

The linker does not allocate the sections automatically.

- Mode1

If any absolute sections exist when allocating sections in a ROM/RAM area specified in the [ROM/RAM Area List], the linker allocates relocatable sections to avoid an overlap to each of the addresses to be allocated. It will properly allocate the sections in a descending order of their alignment values and sizes so that empty area is minimized.

- Mode2

The linker determines whether to allocate sections unspecified for allocation in a ROM area or in a RAM area based on the types of the sections and automatically allocates them in empty area in each area.

The subsequent setting depends how the section is concretely allocated.

Figure 4.5-32 Disposition/Connection Setup Dialog

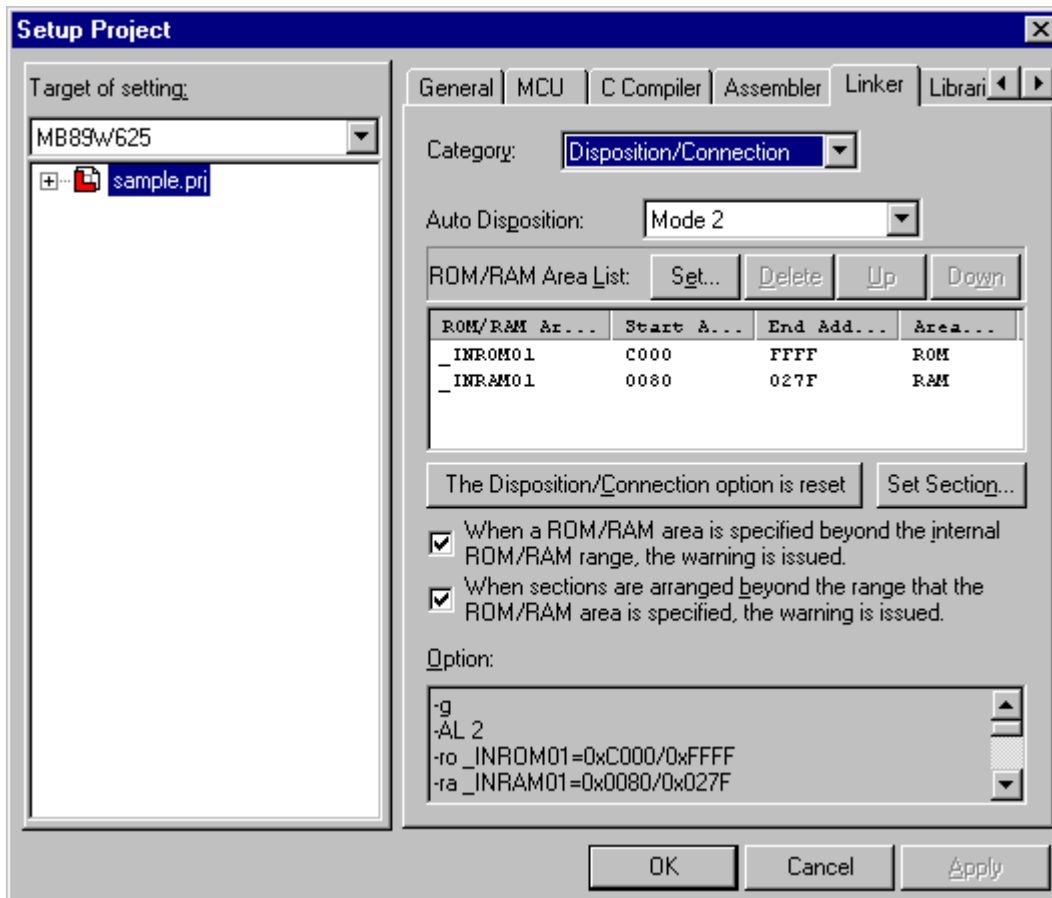
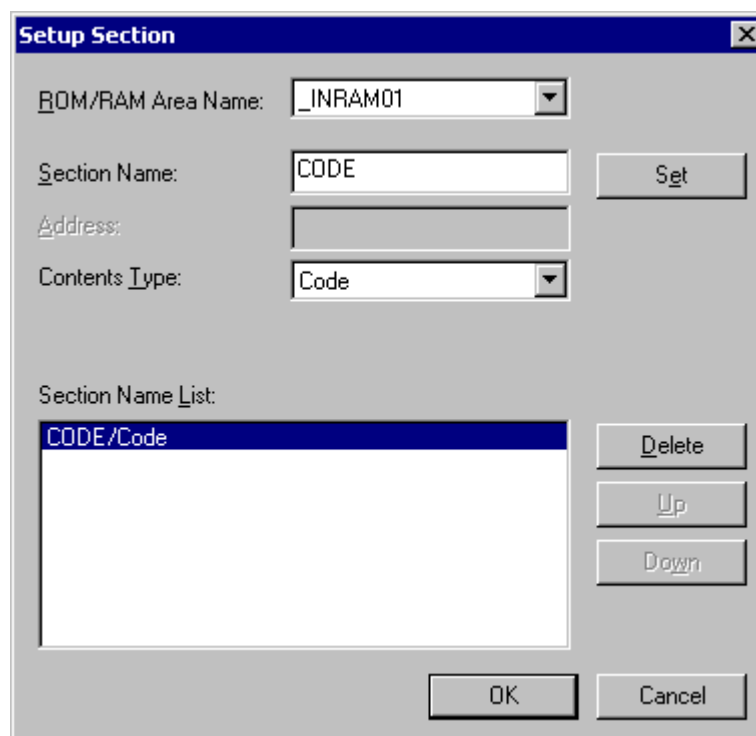


Figure 4.5-33 Setup Section Dialog



■ When a Section is Allocated to the Specified ROM/RAM area

1. Define the ROM/RAM area.
See Defining the ROM/RAM area.
2. Select the ROM/RAM area you want to set from [ROM/RAM Area List].
3. Click the [Set Section...] button.
The section setup dialog shown in Figure 4.5-33 opens.
4. Specify the name of the section to be allocated to the selected area.
A wild card character can be also used.
5. Select a content type from the drop-down list as required.
(None/Code/Data/Stack/Const/IO)
6. Click the [Set] button.
The set section is added to the end of [Section Name List].
7. When all settings are completed, click the [OK] button.

The order in which sections are allocated to the specified ROM/RAM area is the same as the order in [Section Name List]. For how to change this order, see Changing the section allocation order next by one.

■ When a Section is Allocated to the Specified Starting address

1. Click the [Set Section...] button.
The section setup dialog shown in Figure 4.5-33 opens.
2. Select [Specify in Address] from the drop-down list for [ROM/RAM Area Name].
3. Specify the name of the section to be allocated to the ROM/RAM area.
A wild card character can be also used.
4. Specify the starting address to which the section is to be allocated.
5. Select and set a content type from the drop-down list as required.
6. Click the [Set] button.
The specified section is added to the end of [Section Name List].
7. When all settings are completed, click the [OK] button.

To continuously allocate several sections to the specified address, execute step (6), set [Section Name] and [Contents Type] only, then click the [Set] button. Repeat this operation.

Sections are allocated in the specified order. For how to change this order, see the next changing the section allocation order.

■ Changing the Section Allocation Order

1. Click the [Set Section...] button.
The section setup dialog shown in Figure 4.5-33 opens.
2. Specify the area you want to change in [ROM/RAM Area Name]. When an address is directly specified, select [Addressing]. The sections to be allocated to the area (or address) are displayed in [Section Name List].
3. Select the section name whose order is to be changed, then click the [Up] or [Down] button to move the cursor to a relevant position.
For addressing, also see Notes in addressing in the next item.

■ Notes in Addressing

Examples of relationship between descriptions in [Section Name List] and the linker options to be generated are given below. Pay special attention when changing the section allocation order.

[Example 1] When descriptions in [Section Name List] are as follows:

```
sec1=H'1000
sec2
sec3=H'F000
sec4
```

The following sections are generated.

```
-sc sec1+sec2=H'1000
-sc sec3+sec4=H'F000
```

[Example 2] When the sec4 allocation order is changed as follows:

```
sec1=H'1000
sec2
sec4
sec3=H'F000
```

The following sections are generated:

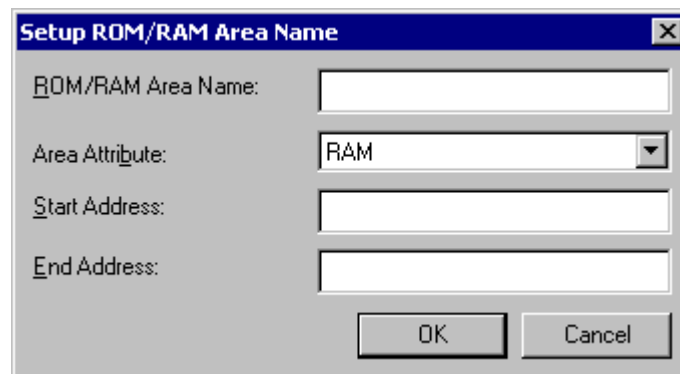
```
- sc sec1+sec2+sec4=H'1000
-sc sec3 =H'F000
```

■ Defining the ROM/RAM area

1. Click the [Add] button.

The Setup ROM/RAM Area Name dialog (Figure 4.5-34) is opened.

Figure 4.5-34 Setup ROM/RAM Area Name Dialog



2. Specify a ROM/RAM area name.
Specify the ROM/RAM name that differs from the set names.
3. Select ROM or RAM as [Area Attribute].
4. Specify a start address and end address (starting address and end address of ROM/RAM area).
5. Click the [OK] button.

The specified ROM/RAM area is registered in [ROM/RAM Area List].

In Auto Disposition (Mode 2), the linker searches an available ROM/RAM area beginning at the top of the [ROM/RAM Area List]. Click the [Up] button or the [Down] button to move to a proper position.

■ Deleting a ROM/RAM area

1. Select the ROM/RAM area you want to delete from [ROM/RAM Area List].
2. Click the [Delete] button.

■ Method of Outputting Warning When ROM/RAM area is Specified Outside Range of Internal ROM/RAM

1. Put a check mark in the checkbox. When a ROM/RAM area is specified beyond the internal ROM/RAM range, the warning is issued.
 - When an area outside the internal-ROM/RAM area is specified in the dialog, a warning dialog is displayed.
 - When an area outside the internal-ROM/RAM area is set, a warning is output at linking.

■ Method of Outputting Warning When Section is Placed Outside Specified ROM/RAM area Range

1. Put a check mark in the checkbox. When sections are arranged beyond the range of specified ROM/RAM area, the warning is issued.
 - When a section is placed outside the specified ROM/RAM area (-ro and -ra options) or outside the internal-ROM/RAM area in the MCU, a warning is output at linking.

■ When Initializing [Placement/Link] Option Based on MCU Information

1. Click the [The Disposition/Connection option is reset] button.
 - The placement/link option is reset.
When this reset is performed, the placement/link option is set as follows:
 - Auto Disposition (-AL 2)
Mode 2 (optimum automatic placement by linker) is set.
 - ROM/RAM area (-ro, -ra)
The ROM/RAM area is all cleared and the currently selected MCU internal ROM/RAM address is set.
 - Section information (-sc)
Section information is all cleared.
 - A warning is output when a ROM/RAM area outside the range of the internal ROM/RAM (-check_rora) is specified.
Setting is performed so as to output a warning. However, when an MCU without single-chip mode is selected, setting is performed so as not to output a warning.
 - A warning is output when a section is placed outside the specified ROM/RAM area range (-check_locate)
Setting is performed so as to output a warning.

4.5.5.7 Setting Librarian Options

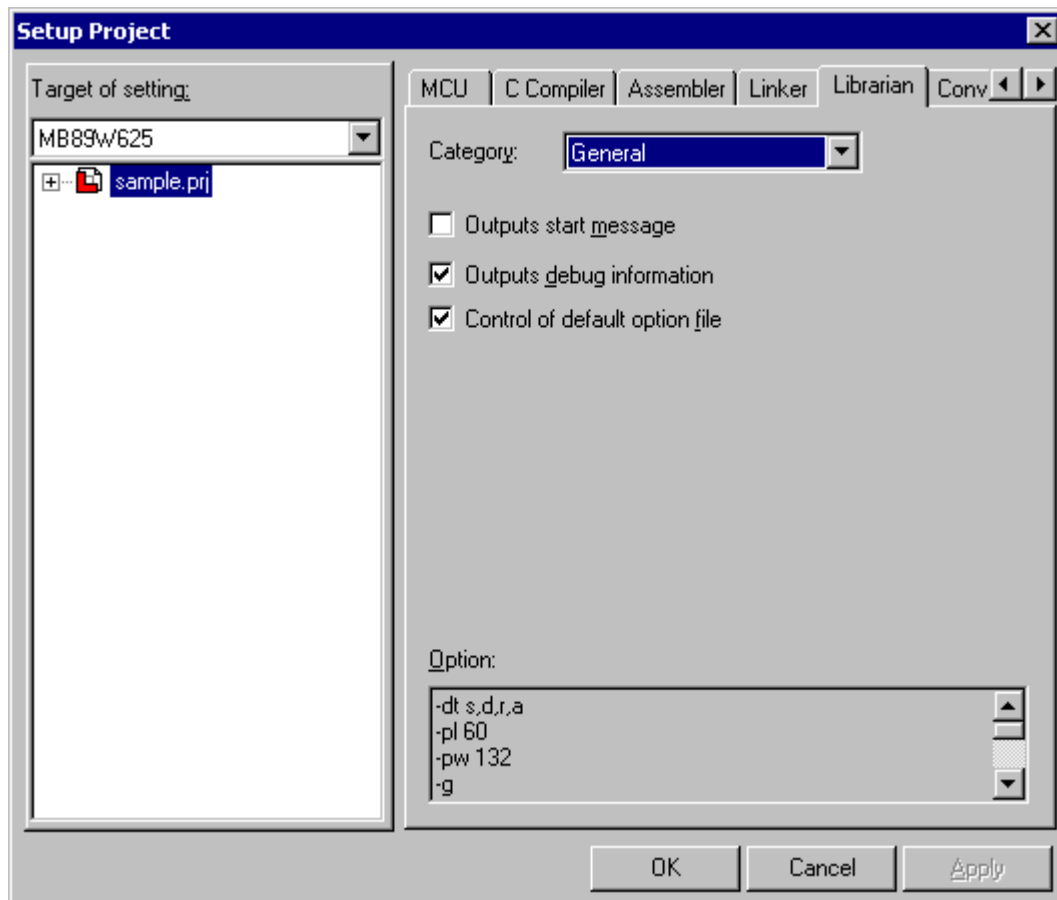
This section explains how to set librarian options.

■ Setting Librarian Options

1. Click [Librarian] tab from the project setup dialog.
2. Select category.
[General] or [Output List] can be selected as category.
Specified options can be checked using [Option] at the bottom of the dialog.

■ Setting Options in [General]

Figure 4.5-35 Librarian Option Setup Dialog (General)

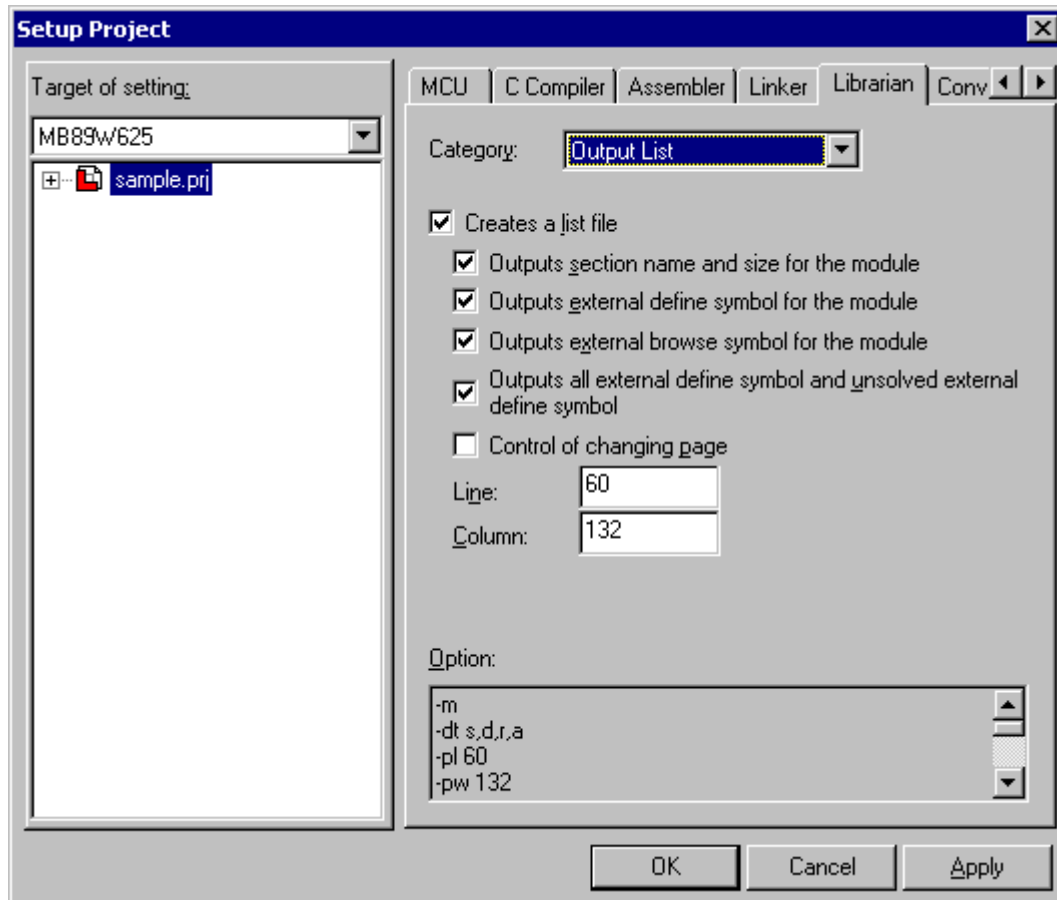


The following options can be set:

- Output start message (-v)
- Output debug information (-g)
- Control of default option file (-Xdof)

■ Setting the Output-listing

Figure 4.5-36 Librarian Option Setup Dialog (Output List)



1. Select any of the following output types:
 - Outputs section name and size for module
 - Outputs external define symbol for module
 - Outputs external browse symbol for module
 - Outputs all external define symbol and unsolved external define symbol
2. To suppress page change, set a check mark to the left of [Control of changing page].
When page change is suppressed, the line count cannot be set.
3. Set Line Count and Column Count to the right of [Line] and [Column] as required.

■ Starting Librarian

If the project type is a "library file," the librarian is started.

To change the project type, see Section "4.5.5.1 General".

4.5.5.8 Setting Converter Options

This section explains how to set converter options.

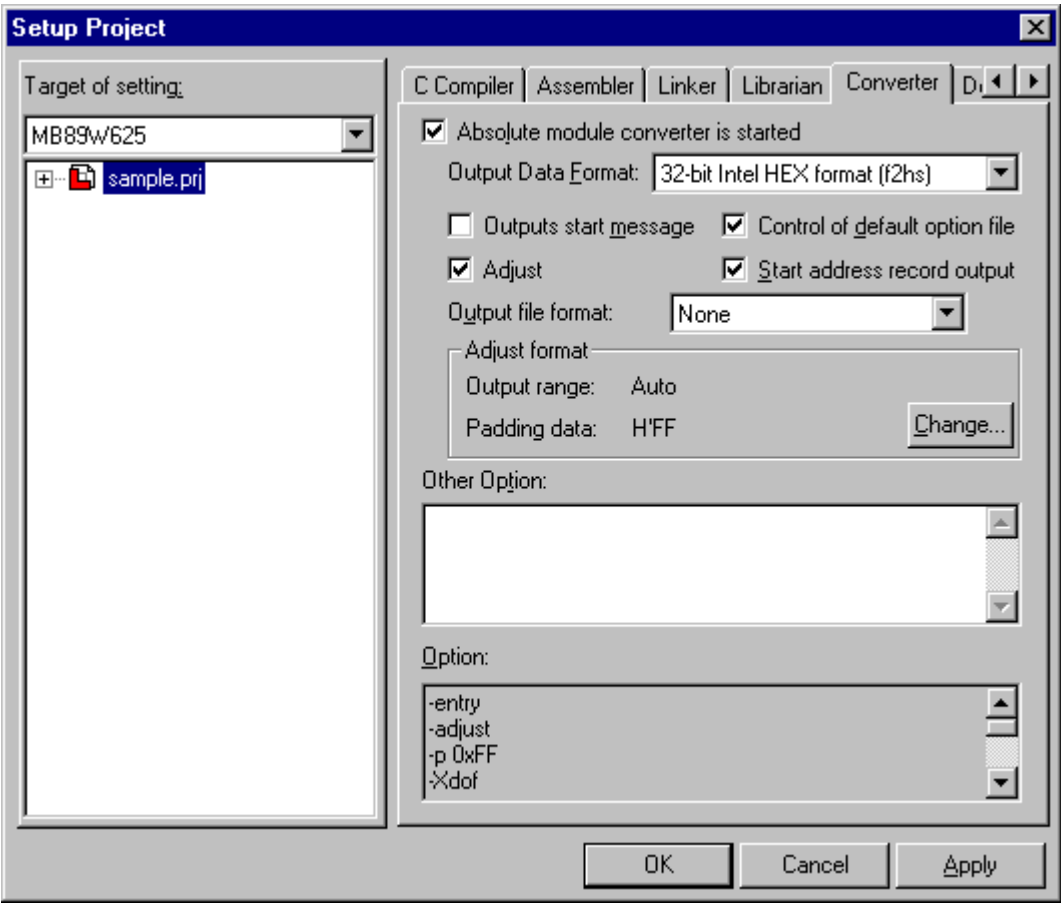
■ Setting Converter Options

1. Click the [Converter] tab from the project setup dialog.
2. To start the load module converter marks the [Start load module converter] checkbox.
3. Select a conversion format.
 - Motorola S format(f2ms)
 - Converts the absolute format load module of linker output to an S format. Data at addresses 0 to 0xFFFFFFFF is converted.
 - 32-bit Intel HEX format(f2hs)
 - Converts the absolute format load module of linker output to an HEX format. Data at addresses 0 to 0xFFFFFFFF is converted.
 - Intel HEX format(f2is)
 - Converts the absolute format load module of linker output to an HEX8 format. Data at addresses 0 to 0xFFFF is converted.
 - This format is left to maintain compatibility with the previous version. The 32-bit Intel HEX format (f2hs) should be used for conversion to an HEX8 format.
 - Intel Extend HEX format(f2es)
 - Converts the absolute format load module of linker output to an HEX16 format. Data at addresses 0 to 0xFFFF is converted.
 - This format is left to maintain compatibility with the previous version. The 32-bit Intel HEX format (f2hs) should be used for conversion to an HEX16 format.
4. Set the following items as required:
 - Common Options
 - Outputs start message (-V)
 - Control of default option file (-Xdof)
 - Options for Motorola S format (f2ms)
 - Output file format - None
 - Outputs data in mixed-S1, S2, and S3 records according to the data address.
 - Output file format - S1 record (16-bit address) (-S1)
 - Outputs data in an S1 record (in an allowable output range of 0x0000 to 0xFFFF).
 - Output file format - S2 record (24-bit address) (-S2)
 - Outputs data in an S2 record (in an allowable output range of 0x000000 to 0xFFFFFFFF).
 - Output file format - S3 record (32-bit address) (-S3)
 - Outputs data in an S3 record (in an allowable output range of 0x00000000 to 0xFFFFFFFF).
 - Adjust (-adjust)
 - Automatically calls the Format Adjuster to adjust a data output format.

- Output range (-ran)
Specifies the range to be adjusted by an address when selecting the option (-adjust) for adjusting an output file. Selecting [Auto] will obtain the starting/ending address for adjustment from the absolute format load module to set automatically.
- Padding data (-p)
When selecting the option (-adjust) for adjusting an output file, the area of the file where no data exists is packed with data having a specified value.
- Options for 32-bit Intel HEX format (f2hs)
 - Output file format - None
Outputs data in mixed-HEX8, HEX16, and HEX32 according to the data address.
 - Output file format - HEX8 (16-bit address) (-I16)
Outputs data in HEX8 format (in an allowable output range of 0x0000 to 0xFFFF).
 - Output file format - HEX16 (20-bit address) (-I20)
Outputs data in HEX16 format (in an allowable output range of 0x00000 to 0xFFFFF).
 - Output file format - HEX32 (32-bit address) (-I32)
Outputs data in HEX32 format (in an allowable output range of 0x00000000 to 0xFFFFFFFF).
- Adjust (-adjust)
Automatically calls the Format Adjuster to adjust a data output format.
- Start address record output (-entry)
Outputs the starting address record. The starting address record will be used as a starting address for loading (at the time of debugging).
- Output range (-ran)
Specifies the range to be adjusted by an address when selecting the option (-adjust) for adjusting an output file. Selecting [Auto] will obtain the starting/ending address for adjustment from the absolute format load module to set automatically.
- Padding data (-p)
When selecting the option (-adjust) for adjusting an output file, the area of the file where no data exists is packed with data having a specified value.

The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE Workbench User's Manual

Figure 4.5-37 Converter Option Setup Dialog



4.5.5.9 Setting Debug Options

This section explains how to set debug options.

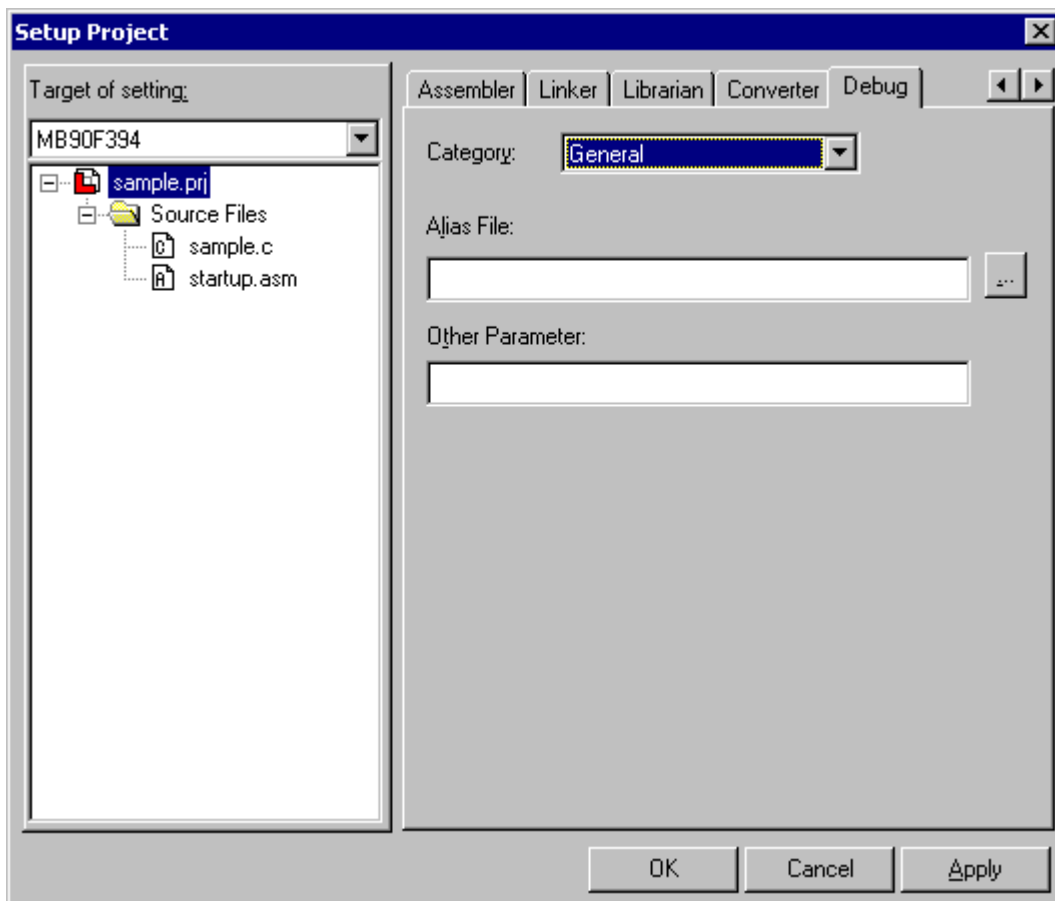
■ Setting Debug Options

1. Click the [Debug] tab from the project setup dialog.
The debug option setup dialog shown in Figure 4.5-38 opens.
2. Select category.
[General] or [Setup] can be selected as category.

■ Setting Options in [General]

1. Set an alias file.
Clicking the [Browse] button to the right of the input field enables file reference.
2. When setting other options, write them in [Other Parameter].

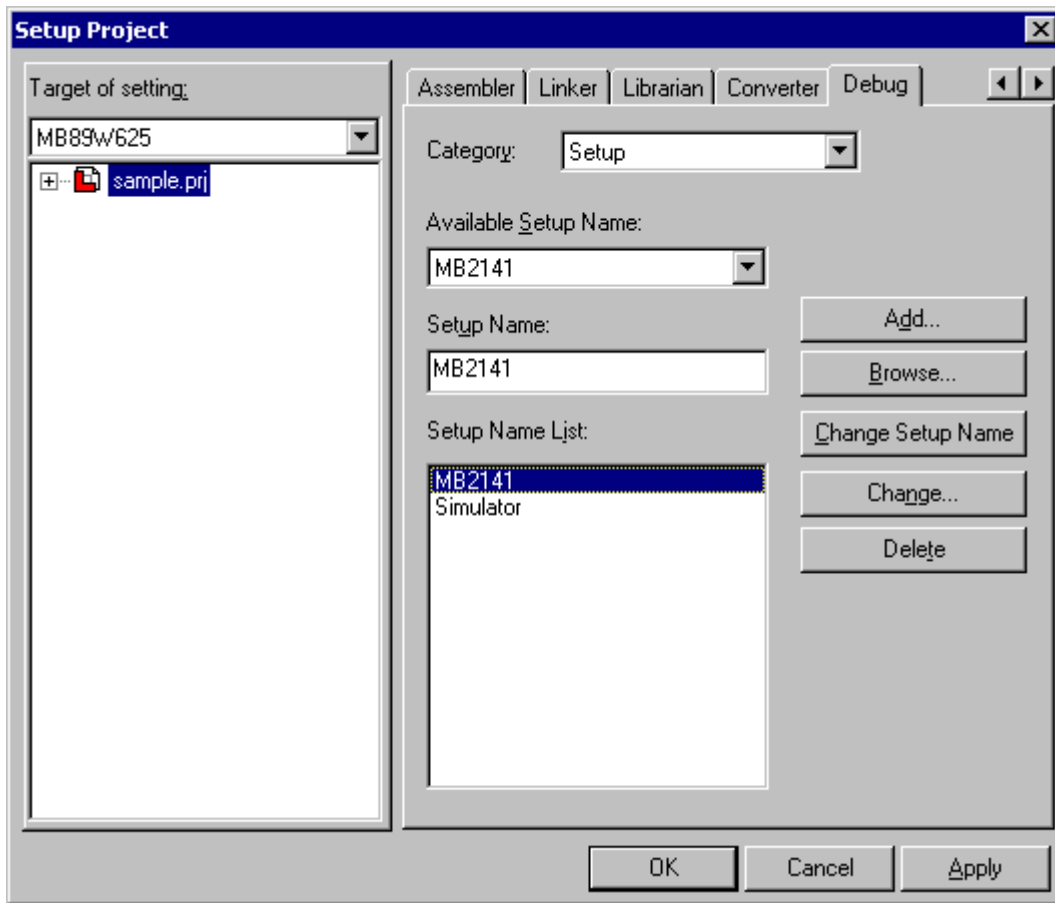
Figure 4.5-38 Debug Option Setup Dialog



■ Setting Options in [Setup]

1. Click the [Debug] tab from the project setup dialog.
2. Select [Setup] category.

3. Set [Setup Name].
Set a different name from registered setup names.
4. Click the [Add] or [Browse] button.
Clicking the [Add] button starts the Setup Wizard and adds the new setup. See Section "4.7.2.4 Setup Wizard".
Clicking the [Browse] button reads information from the set file for setup. When the file selection dialog opens, select a file from the dialog, then click the [Open] button.

Figure 4.5-39 Debug Option Setup Dialog

■ Deleting Debugger Setup

1. Click [Debug] tab from the project setup dialog.
2. Select [Setup] category.
3. Select the setup name to be deleted from [Setup Name List].
4. Click the [Delete] button.

■ Changing Debugger Setup

1. Click [Debug] tab from the project setup dialog.
2. Select [Setup] category.
3. Select the setup name to be changed from [Setup Name List].
4. Click the [Change] button.

Setup Wizard is started. See Section "4.7.2.4 Setup Wizard".

■ Changing Setup Name

1. Click [Debug] tab from the project setup dialog.
2. Select [Setup] category.
3. Select the setup name to be changed from [Setup Name List].
4. Click the [Change Setup Name] button.

4.5.6 Setting Customize Build

This section explains how to start a different tool before or after executing the language tool during Make or Build.

■ Customize Build Function

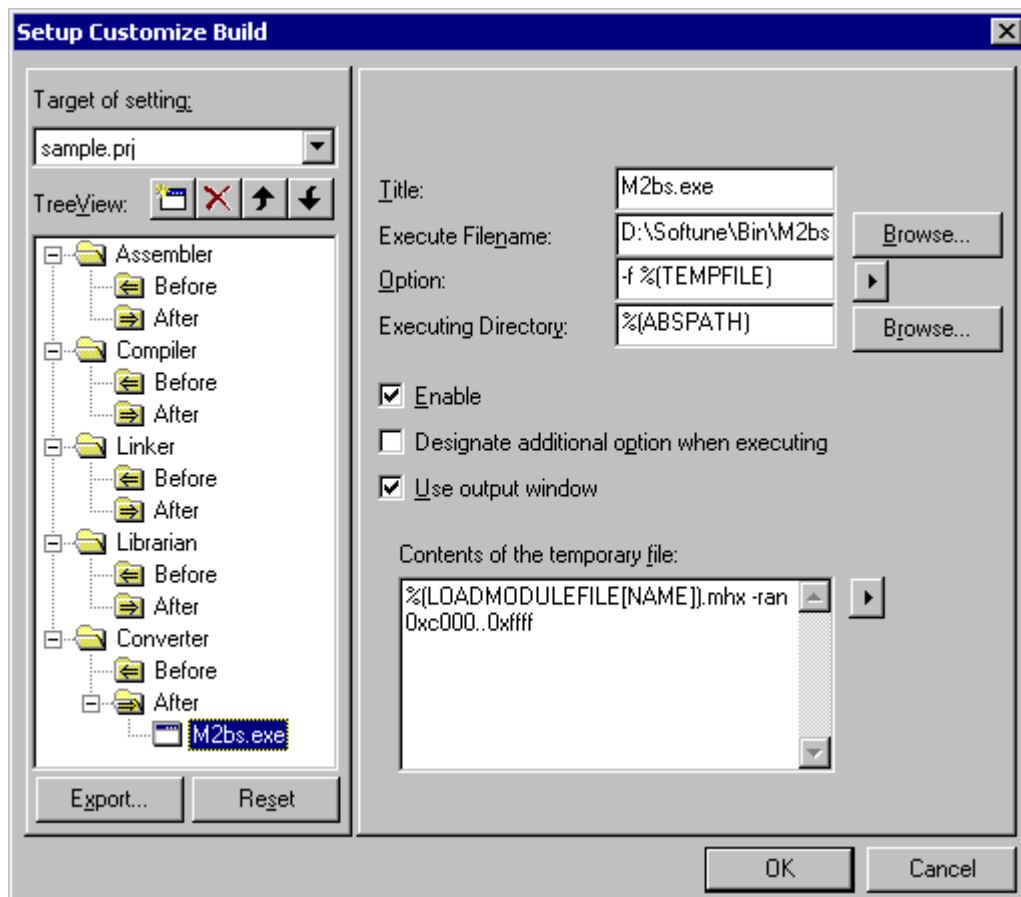
In SOFTUNE Workbench, it is possible to make a different tool operate automatically before or after executing the language tool during Compile, Assemble, Make, or Build. Using this function makes it possible to:

- make a customer tool operate before the Compiler is executed.





- make the object module conversion tool operate after the Linker is executed.

This setting is stored on a project-by-project basis.

Figure 4.5-40 Setup Customize Build Dialog



■ Tool Button List

	New
	Delete
	Up
	Down

■ Setting Target

There are two types of setting targets in [Target of setting]: Default and Project name.

See [Target of setting] for which one is currently set.

- Default

When the Customize Build is set when no project is opened, the default setting can be changed.

The Customize Build setting is referred when creating a new project and is copied to that project.

"Default" is displayed in [Target of setting].

Note:

When a project created in an older version is opened, this setting is also referred and copied to that project.

- Project Name

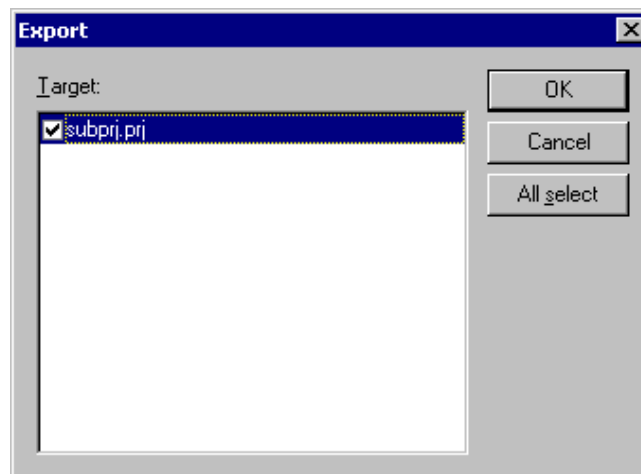
Customize build is set for the project. The tool operates at Compile, Assemble, Make, or Build for the opened project.

■ Export

The [Export] button is clicked to open the Export dialog shown in Figure 4.5-41.

By specifying the project in this dialog, tool information can be copied to other projects in workspace.

Figure 4.5-41 Export Dialog



■ Reset

To clear the currently set state and return to the default setting, click [Reset] button. The [Reset] button can be used when the setting target is Project.

■ Title

Input the tool name; duplicated tool names do not cause a problem.

■ Execution File Name

Input the file name of the executed tool.

■ Option

Specify the option for the executed tool. A macro can be specified in this field.

For the macros, refer Section "1.11 Macro Description Usable in Manager" in the SOFTUNE Workbench User's Manual.

When the button at the right of this field is clicked, the list of usable macros is displayed. For example, when [Build File]-[Directory] is clicked, %(FILE[PATH]) is inserted for the option at the cursor position.

■ Executing Directory

To execute the tool for a particular directory, specify the Executing Directory. If the Executing Directory is not specified, the tool is executed for the project directory.

■ Enable

This specifies whether or not to execute the tool at Compile, Assemble, Make, or Build. The tool is not executed when the checkbox is unchecked. To suspend execution of the tool, uncheck the checkbox.

■ Designate additional Option When Executing

This specifies whether or not to display a dialog in which additional options can be specified at executing the tool. When adding options at executing the tool, check the checkbox.

The option set in the specifying other parameter dialog is added as is to the end of the character string specified for [Option].

■ Use Output Window

To display the tool execution result in the Output window, check the checkbox. There are some precautions to follow when using the Output window. For details, refer to Section "1.11 Macro Description Usable in Manager" in SOFTUNE Workbench User's Manual.

■ Contents of the Temporary File

When the %(TEMPFILE) macro is specified for [Option], SOFTUNE Workbench creates a new temporary file at executing the tool and deletes it at ending the tool execution. In this field, specify the data to be written to this temporary file. For example, when the tool option becomes very long, it is possible to specify %(TEMPFILE) for [Option] and specify that option in this field (However, this is only valid when the tool to be executed permits specifying the option in the file.)

Macros can be input in this field. For the macros, refer to Section "1.11 Macro Description Usable in Manager" in SOFTUNE Workbench User's Manual.

■ Setting Procedure

1. Select [Project]-[Customize Build].
 - The Customize Build setting dialog is displayed (Figure 4.5-40).
2. Select the tool registered position from the tree view and then press the [NEW] button.
 - When a category is selected and the [NEW] button is pressed, a tool entitled "NEWTOL" is created at the end of the category. When a tool is selected and the [NEW] button is pressed, a tool entitled "NEWTOL" is created immediately after the selected tool.
 - Tools in the Before/After category are executed sequentially from the top.
3. Set the title of the tool to be registered.

4. Set the execution file name to be registered.
 - When the [Browse] button at the right of this field is clicked, the file selection dialog is displayed and the tool execution file name can be selected from this dialog.
5. Set the option if necessary.
6. Set the execution-time directory if necessary.
 - When the [Browse] button at the right of this field is clicked, the directory selection dialog is displayed and the execution-time directory can be selected from this dialog.
7. Set [Enable], [Designate additional option when executing], and [Use Output window] if necessary.
 - Select a category or two or more tools from the tree view to batch-change [Enable], [Designate additional option when executing], and [Use Output window].
8. Set the temporary file data as necessary.
9. Click the [OK] button to complete setting.

■ Deletion Procedure

1. Select [Project]-[Customize Build].
 - The Customize Build setting dialog is displayed (Figure 4.5-40).
2. Use the tree view to select the title of the tool to delete.
 - When a category is selected, all the tools in the category are deleted.
 - Two or more tools can be also selected.
3. Click the [DELETE] button.

■ Start Sequence Change Procedure

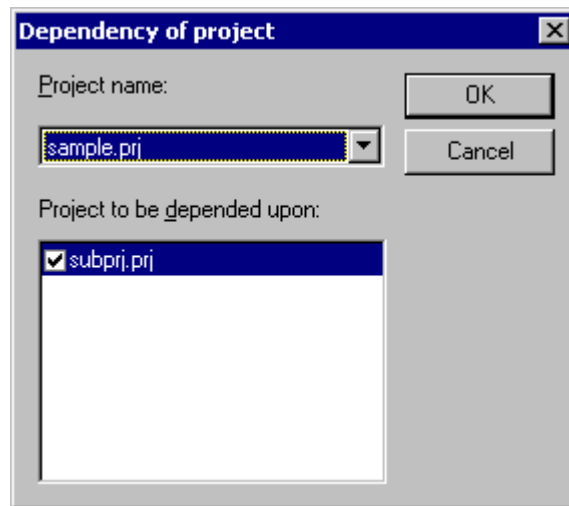
1. Select [Project]-[Customize Build].
 - The Customize Build setting dialog is displayed (Figure 4.5-40).
2. Use the tree view to select the tools for which the start sequence is to be changed.
 - Tools in the Before/After category are sequentially executed from the top.
3. Click the [UP] and [DOWN] buttons to arrange in the start sequence.
 - Tools can be moved only within the category which the tools belong.

4.5.7 Project Dependencies

A subproject is defined in the project.

■ Project Dependencies

Figure 4.5-42 Dependency of Project Dialog



- Project name

The name of the project, which a subproject is defined in or deleted from, is displayed.

- Project to be depended upon.

The name of the project on which the selected project can depend in "Project name" is displayed.

The project name indicated by checkmark is the name of the subproject in the project selected in "Project name".

■ Procedure for Defining Project Dependence

For the procedure to define project dependence, see Section "2.6 Definition of Subproject".

4.5.8 Project Configuration

The project configuration is set.

■ Project Configuration

There are the following menus to set the project configuration:

- Add and Delete

The project configuration is added and deleted, and the active configuration is changed.

- Configuration at build

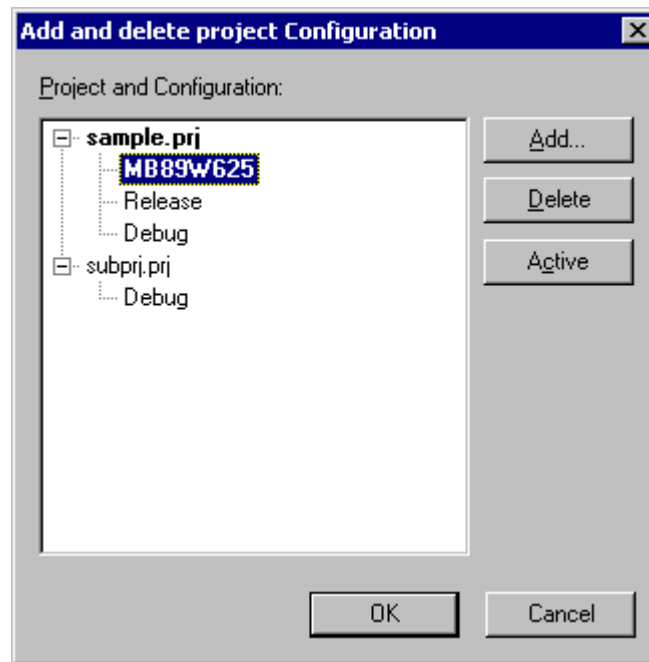
The configuration as a subproject is made or built is set.

4.5.8.1 Project Configuration - Add and Delete

The project configuration is set (Add and Delete).

■ Project Configuration - Add and Delete

Figure 4.5-43 Add and Delete Project Configuration Dialog



- Project and Configuration

All projects in workspace and their configurations are displayed.

- Add

Click this button to open the [Add Project Configuration] dialog shown in Figure 4.5-44.

This dialog enables the addition of the project configuration.

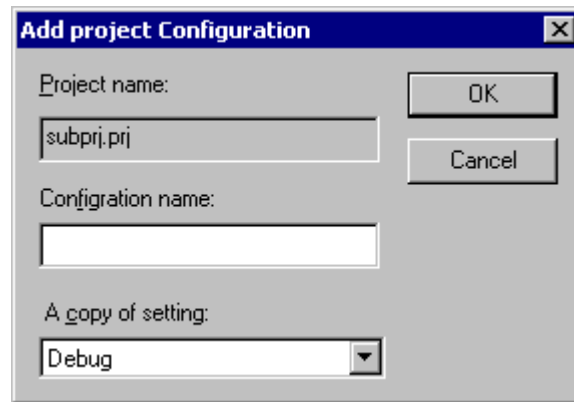
- Delete

Click this button to delete the selected project and its configuration.

- Active

Click this button to make the selected project and its configuration active.

Figure 4.5-44 Add Project Configuration Dialog



- Project name

The name of the project to which the configuration is added is displayed.

- Configuration name

Set the name of the project configuration to be added.

- A copy of setting

Select the configuration to which settings are copied.

■ Setting Procedure

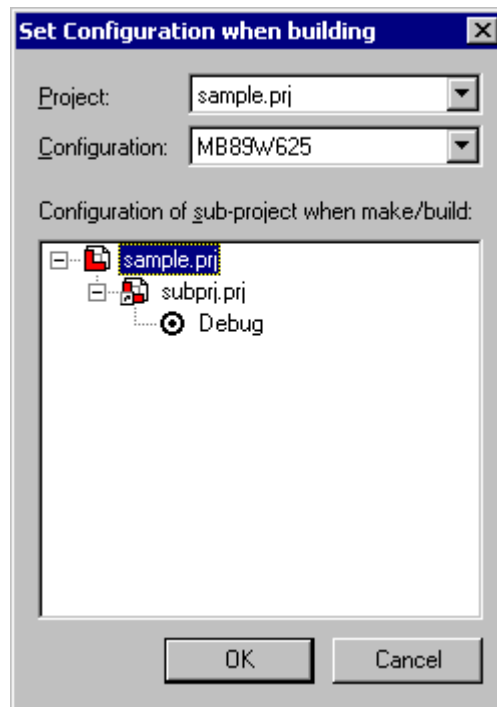
For the setting procedure, see Section "2.7 Creation of Project Configuration".

4.5.8.2 Project Configuration - Configuration at Build

The configuration as a subproject is made or built is set.

■ Project Configuration - Configuration at Build

Figure 4.5-45 Set Configuration When Building Dialog



- Project

Select the parent project of a subproject.

- Configuration

Select the configuration of the parent project selected in "Project."

- Configuration of sub-project when make/build

Select the configuration of the subproject, as the configuration of the parent project selected in "Configuration" is made or built.

■ Setting Procedure

For the setting procedure, see Section "2.10.1 Making or Building of Project".

4.5.9 Include Dependencies

"Include Dependencies" updates include file dependency.

■ Include Dependencies

This command (function) checks all the source files in the project file and registers all the include files being used by the source files in the project. The registered include files are displayed in the [Dependencies] category field of the SRC tab of project Window.

4.5.10 Compile, Make, Build, and Stop

This section explains the functions of Compile, Make, Build, and Stop.

■ Compile

Compile compiles only the specified source file irrespective of whether other source files and include files are corrected. However, compile does not link the specified source file.

This command also assembles the specified file when the file is an assembler source file.

■ Make

Make checks all the source and include files in the project and compiles or assembles only files which have been modified since the last Make or Build.

If some library and object files are modified, make links them to create a target file.

■ Build

Build compiles or assembles all the source files in the project regardless if they have been modified since the last Make or Build.

It also links all object and library files to create a target file.

■ Stop

Use Stop when you want to stop compile, make, or build execution for some reason.

■ Checking Source File Saving

When an unsaved file is being edited by the standard editor, execute compile (assemble), make, or build, then save the file. If a check mark is set to the left of [Inquiry for Save at Compile/Assemble] in [Setup]-[Development...]-[Project], however, the dialog for asking whether to save the file opens. When the [No] button is clicked, the source file is compiled without being saved (Source file before editing).

4.6 Debug

"Debug" starts and terminates debugging and controls the debugger when SOFTUNE Workbench is in the debug session.

■ Debug Start and Termination

- Loading Target File
- Start Debug/End Debug

■ Debugger Control When SOFTUNE Workbench is in the Debug Session

- Run
- Abort
- Reset MCU
- Break Point
- Break Point Set/Reset
- Event
- Sequence
- Stack
- Time Measurement
- Call
- Clear Call
- Vector

4.6.1 Run

This section explains the debugger program execution function.

■ Run

"Run" provides the following five functions:

-Go

When [Go] is clicked, the debugger continuously executes the program from the current PC position. When a break point is reached or when [Abort] is selected from the [Debug] menu, the debugger stops program execution.

-Step In

When [Step In] is clicked, the debugger executes the step, moves the PC to the address of the next instruction, and stops. When a function call instruction is executed, the debugger stops at the beginning of the function.

-Step Over

When [Step Over] is clicked, the debugger executes the step, moves the PC to the beginning of the next instruction, and stops. When a function call instruction is executed, the debugger executes all the functions, moves the PC to the next instruction address of the function call instruction, and stops.

-Step Out

When [Step Out] is clicked, the debugger executes the current function to the end, returns control to the function caller, moves the PC to the next instruction address of the function call instruction, and stops.

-Run Until Cursor

When [Run Until Cursor] is clicked, the debugger executes the program to the instruction immediately before the address indicated by the cursor (in the Source or Assembly Window), moves the PC to the address, and stops.

4.6.2 Abort

This section explains the debugger program execution stop function.

■ Abort

[Abort] is used to forcibly interrupt the program being executed by the debugger. When the program stops, the PC moves to the next instruction address of the last executed instruction. Source line display and disassemble display are also updated according to the PC value set when the program stopped.

4.6.3 Reset MCU

This section explains the MCU reset function of the debugger.

■ MCU Reset Function

The MCU reset function resets the MCU.

- Emulator debugger

The MCU reset function issues the reset signal to the emulator.

- Simulator debugger

As with the actual chip, set the initial values of the registers to be initialized by reset and clear other registers to 0.

In reset MCU, break points, watch points, map setting, and program variables are not modified.

4.6.4 Break Point

This section explains how to set, reset, and modify a break point and how to display a break list.

■ Break Point

When the PC passes an address or the program accesses data at an address, the position where program execution by the debugger is to be stopped is called a break point.

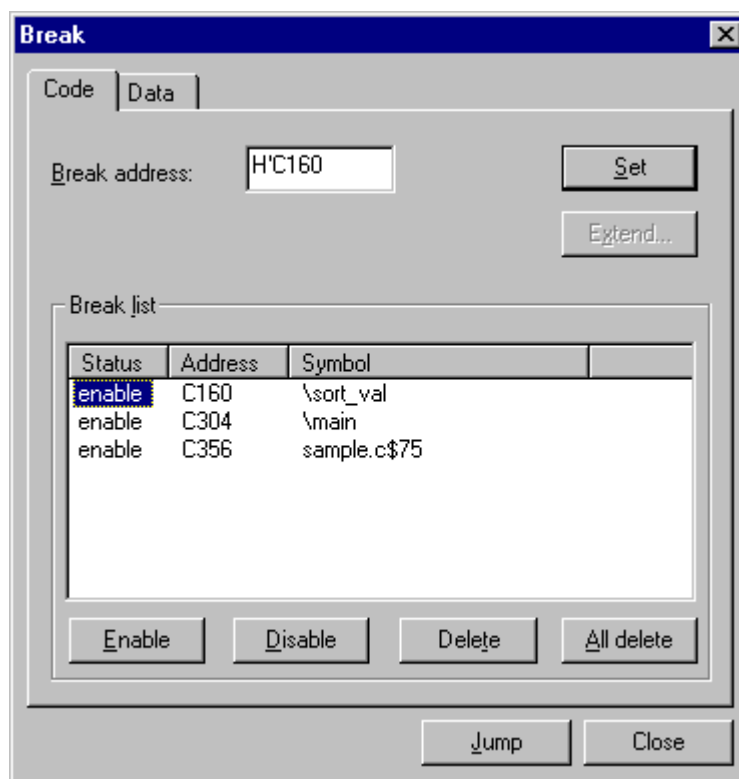
■ Code Break Point

When the PC passes the set address (when the address is executed), the break point where program execution is to be stopped is called a code break point.

■ Setting of Code Break Point

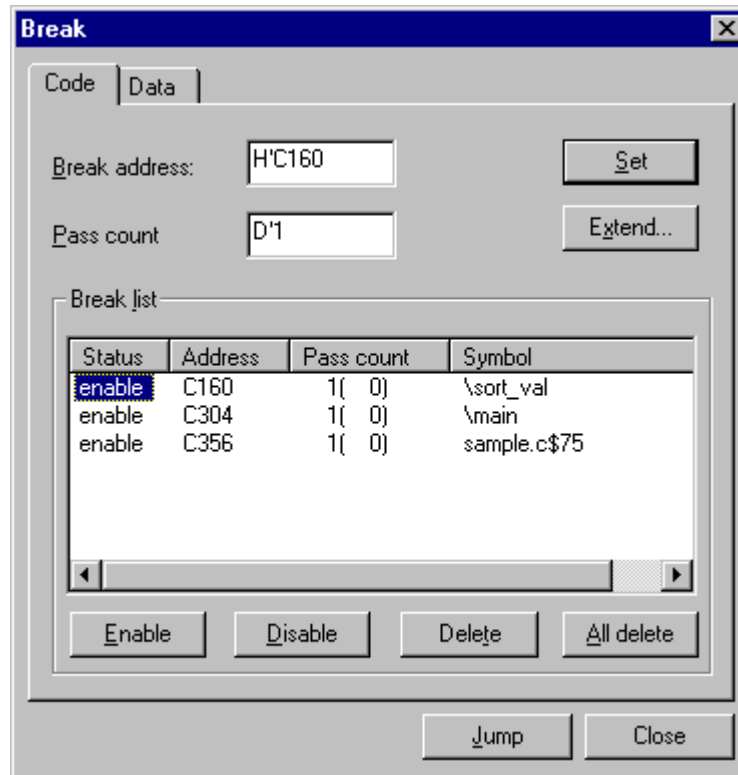
[Emulator debugger]

Figure 4.6-1 Break Dialog (Code)



[Simulator debugger]

Figure 4.6-2 Break Dialog (Code)



- Break address

In this field, the address that sets a break point is specified.

- Pass count

In this field, the count of times the PC passes that point (a particular point) before causing a break is set. This field is only enabled for the simulator debugger.

- Break list

A list of code break points currently being set is displayed.

State : Enable or disable is displayed.

Address : The set address is displayed.

Pass count : The set pass count is displayed. In (), the count of times the PC has passed that point by the present time is displayed.

Symbol : The symbol given to that address is displayed.

- [Set] button

These buttons are used to set a break point at the specified address. When an address that is already set in the break list is specified, the set data of the break point at this address is changed.

- [Enable] button

This button is used to enable the break points in the break list currently being selected.

- [Disable] button

This button is used to disable the break points in the break list currently being selected. The break points are simply disabled; that is, the setting itself of the break points is not canceled.

- [Delete] button

This button is used to delete the setting of the break points in the break list currently being selected.

- [All Delete] button

This button is used to delete the setting of all the break points in the break list.

- [Jump] button

Moves a starting position for displaying the source window to the code position at the break point selected from the break list.

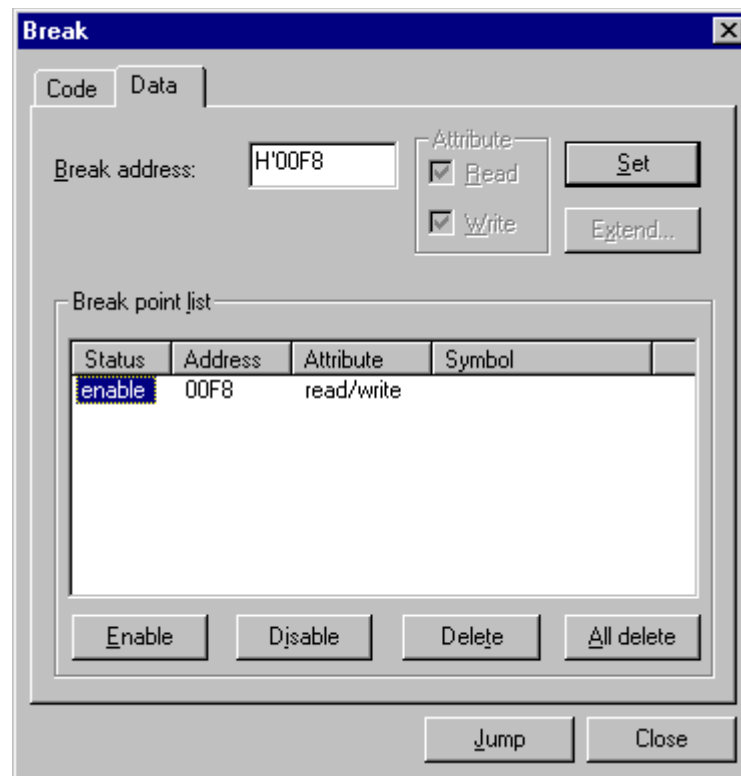
■ Data Break Point

Such a break point as stop the program when data at the set address is accessed is called the data break point.

■ Setting of Data Break Point

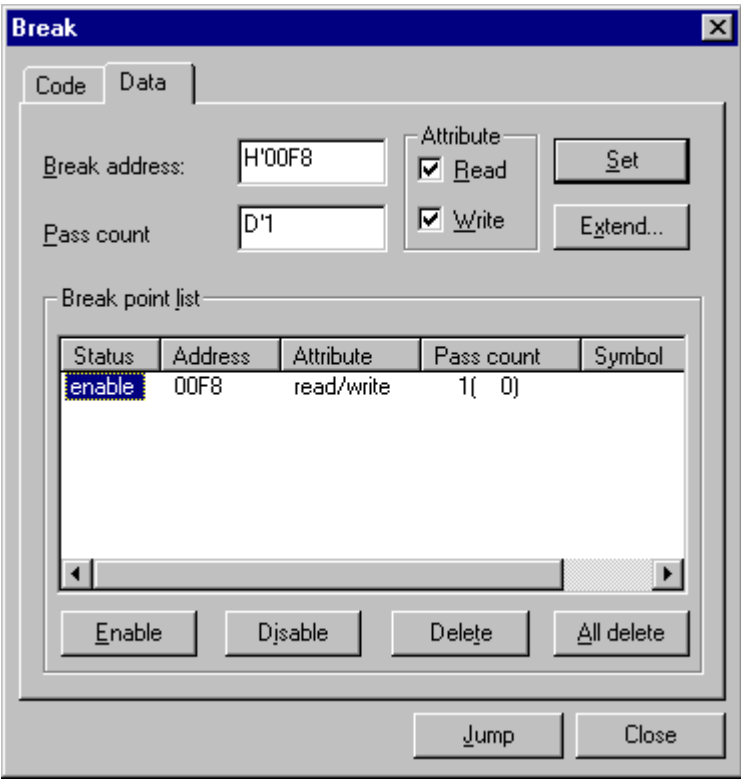
[Emulator debugger]

Figure 4.6-3 Break Dialog (Data)



[Simulator debugger]

Figure 4.6-4 Break Dialog (Data)



- Break point address
In this field, the address that sets a break point is specified.
- Pass count
In this field, the count of times access with the specified attribute is to be made to that point (a particular point) before causing a break is set. This field is only enabled for the simulator debugger.
- Attribute
In this field, read access, write access, or both is specified.
- Break point list
A list of data break points currently being set is displayed.
Status : Enable or disable is displayed.
Address : The set address is displayed.
Attribute : The attribute of the break point is displayed.
Pass count : The set pass count is displayed. In (), the count of times the PC has passed that point by the present time is displayed.
Symbol : The symbol given to that address is displayed.
- [Set] button
These buttons are used to set a break point at the specified address. When an address that has been already set in the break point list is specified, the set data of the break point at this address is changed.

- [Details] button

This button is used to set the details of a break point at the specified address. For details, see Section "4.6.4.1 Break Point Details Setting".

- [Enable] button

This button is used to enable the break points in the break point list currently being selected.

- [Disable] button

This button is used to disable the break points in the break point list currently being selected. The break points are simply disabled; that is, the setting itself of the break points is not canceled.

- [Delete] button

This button is used to delete the setting of the break points in the break point list currently being selected.

- [All Delete] button

This button is used to delete the setting of all the break points in the break point list.

- [Jump] button

Displays data at the break point selected from the break list in the memory window.

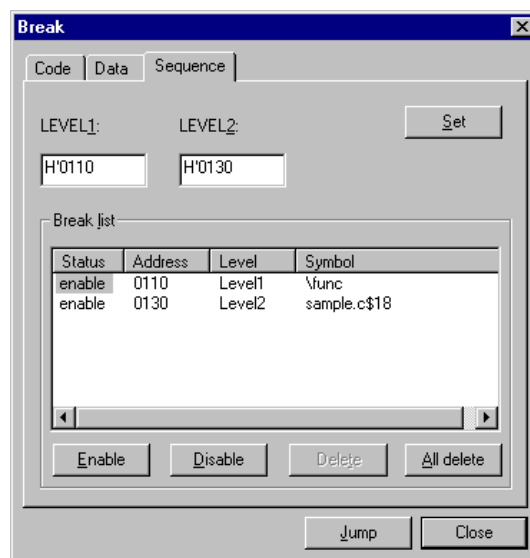
■ Sequence Break Points

When two set addresses are passed the order of LEVEL1 → LEVEL2, the break points to be stopped the program is called a sequence break points.

■ Setting of Sequence Break Points

[Emulator (MB2146-09)]

Figure 4.6-5 Break Dialog (Sequence)



- LEVEL

In this field, the address that sets a break point is specified.

- Break list

A list of sequence break points currently being set is displayed

State : Enable or disable is displayed.

Address : The set address is displayed.

LEVEL : The order of address to be the break point condition is displayed.

Symbol : The symbol given to that address is displayed.

- [Set] button

These buttons are used to set a break point at the specified address. When an address that has been already set in the break list is specified, the set data of the break point at this address is changed.

- [Enable] button

This button is used to enable the break points in the break list currently being selected.

- [Disable] button

This button is used to disable the break points in the break list currently being selected. The break points are simply disabled; that is, the setting itself of the break points is not canceled.

- [Delete] button

This button is used to delete the setting of the break points in the break list currently being selected.

- [All Delete] button

This button is used to delete the setting of all the break points in the break list.

■ Setting and Resetting a Break Point

In the Source or Assembly Window, a break point can be easily set at the address indicated by the cursor. The break point set at the address indicated by the cursor can be also reset easily. Set and reset this break point as follows:

- Select [Break Point Set/Reset] from the [Debug] menu.

The break point is alternately set and reset each time [Break Point Set/Reset] is selected.

- Click the left button of the mouse in the break point display field of each window.

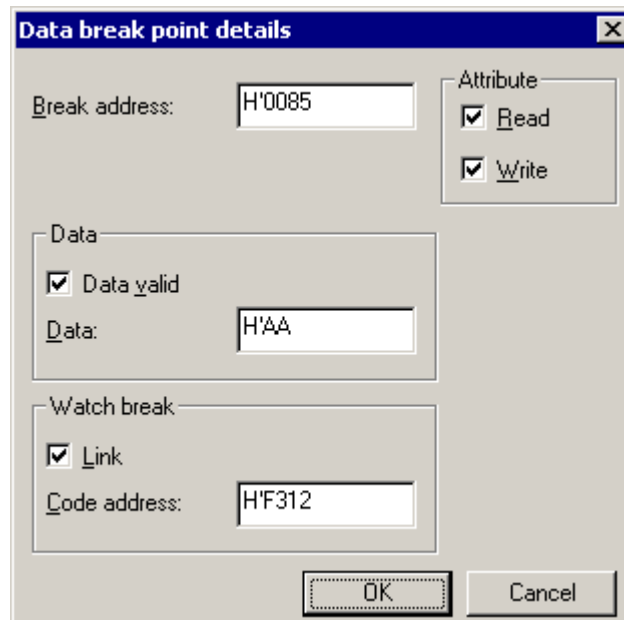
The break point is alternately set and reset each time the left button is clicked.

4.6.4.1 Break Point Details Setting

Setting of break point details is explained here.

■ Details of Data Break Point: (For MB2146-09)

Figure 4.6-6 Data Break Point Details Dialog



- Break address

In this field, the address that sets a break point is specified.

- Attribute

The attribute at the time of data access is specified.

- Data valid

This checks when the data watch is specified in the break point condition.

- Data

In this field, data at the time of data access is specified.

- Link

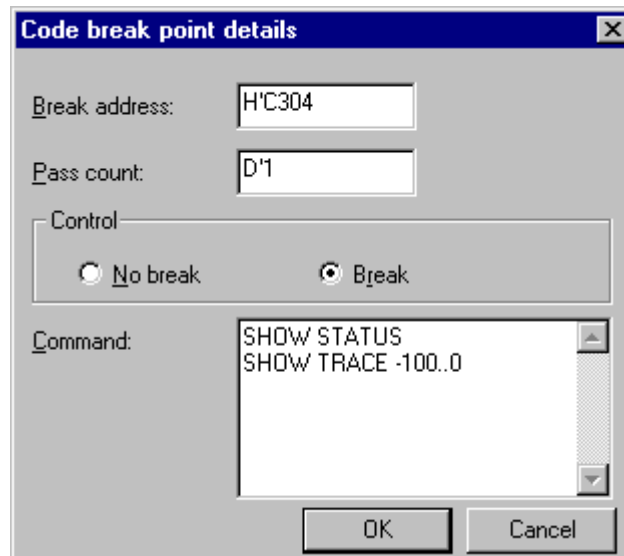
This checks when the link is used as the data monitoring break point.

- Code address

In this field, the break point address of the data monitoring break point is specified.

■ Details of Code Break Point (Simulator Debugger)

Figure 4.6-7 Code Break Point Details Dialog (Simulator Debugger)



- Break address

This field specifies the address to which set a break point.

- Pass count

This field specifies how many times PC needs to pass there to break.

- Control

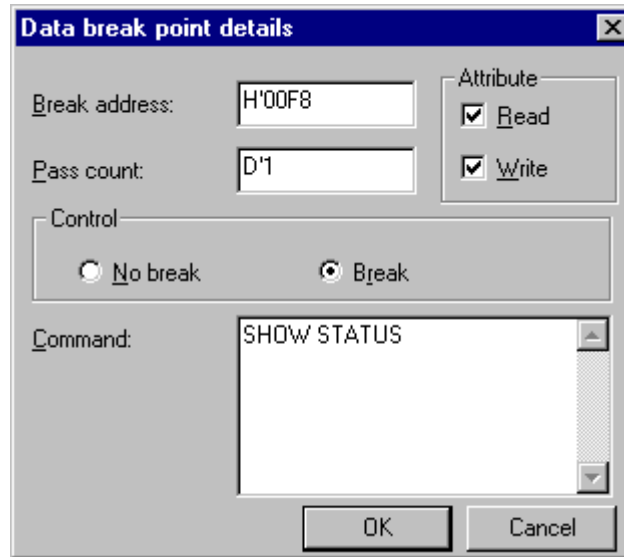
Specifies whether to re-execute or stop execution after the command processing when the break point is hit.

- Command

Specifies the command line to execute when the break point is hit.

■ Details of Data Break Point (Simulator Debugger)

Figure 4.6-8 Data Break Point Details Dialog (Simulator Debugger)



- Break address

This field specifies the address to which set a break point.

- Pass count

This field specifies how many times PC needs to pass there to break.

- Attribute

Specifies the attributes when data accessing.

- Control

Specifies whether to re-execute or stop execution after the command processing when the break point is hit.

- Command

Specifies the command line to execute when the break point is hit.

Note:

For more information about STUB function, refer to Section "2.1.7 STUB Function" in SOFTUNE Workbench User's Manual.

4.6.5 Event

This section explains how to set SOFTUNE Workbench events.

■ Setting Events

Events can be set from the event dialog shown in Figure 4.6-9. This function can be used only in the emulator (MB2141).

[MB2141 (event mode: Normal)]

Figure 4.6-9 Event Dialog (Event)

The Event Dialog (Event) window is shown. It contains the following fields and controls:

- Event number:** A dropdown menu currently showing '1'. A 'Set' button is located to its right.
- Address:** A text box containing 'H'0000'. To its right is the text 'No symbol'.
- Address mask:** An empty text box.
- Data:** An empty text box.
- Data mask:** An empty text box.
- Access attribute:** A dropdown menu currently showing 'Code'.
- External probe data:** An empty text box.
- External probe data:** An empty text box. To its right is a checkbox labeled 'Not specified for data'.
- Event list:** A table with the following header: 'no. en/dis stat addr a_msk data d_msk ep ep_msk sy...'. The table body is currently empty.
- Buttons:** 'Enable', 'Disable', 'Delete', 'All delete', and 'Close'.

- Event Number

Specifies an event number (1 to 8).

- Address

Specifies the address at which the event occurrence condition is to be set.

- Address Mask

Specifies address mask. Only the addresses whose bits are 1 are to be compared.

CHAPTER 4 MENUS

- Data
Specifies the data to be set as the event occurrence condition.
- Data Mask
Specifies data mask. Only the data items whose bits are 1 are to be compared.
- Access Attribute
Specifies a data access attribute. (Code/Read/Write/Read Code/Read Write/Modify)
- External Probe Data
Specifies the external probe data value to be set as the event occurrence condition.
- External Probe Data Mask
Specifies external probe data mask. Only the data items whose bits are 1 are to be compared.
- Not specified for data
Specifies the condition when the data values do not match.
- Event List
Displays the current event setting state.

4.6.6 Sequence

"Sequence" displays the Sequence Window.

■ Sequence

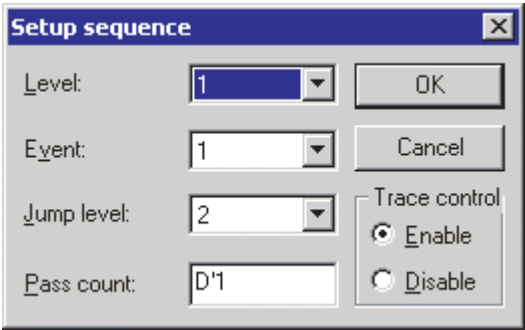
The window that displays the sequence setting state opens. A sequence, latch, and delay count can be set from this window.

This function can be used in the emulator (MB2141).

■ Setting the Sequence

Select [Setup] from the short-cut menu of the Sequence Window.

Figure 4.6-10 Sequence Setting Dialog

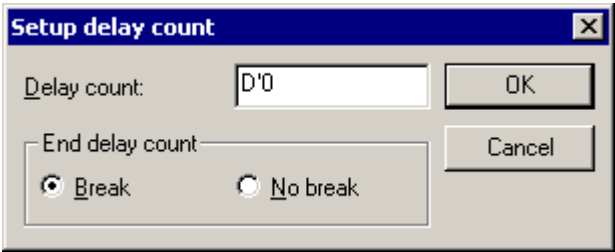


- Level Select a level.(1-8)
- Event Select an event.(1-8/Timer)
- Jump Level Select a jump destination level. (1-8/END)
- Pass Count Set a pass count.
- Trace Control Select "Enable" or "Disable".

■ Setting the Delay Count

Select [Delay Count] from the short-cut menu of the Sequence Window.

Figure 4.6-11 Delay Count Setup Dialog

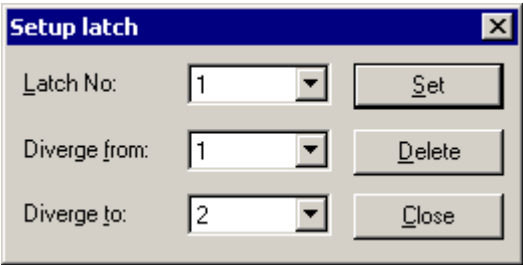


- Delay Count Set the delay count.
- Delay Count Break Select "Break" or "Not Break".

■ **Setting Latch**

Select [Setup Latch] from the short-cut menu of the Sequence Window.

Figure 4.6-12 Latch Setup Dialog

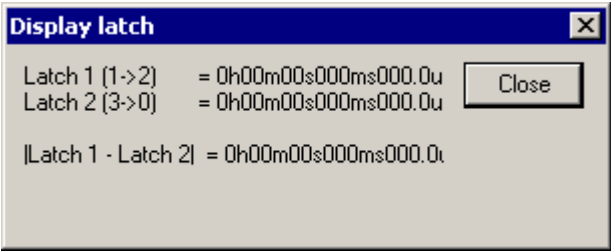


- | | |
|---------------|---|
| -Latch No | Select a latch number. (1-2) |
| -Diverge from | Select a branching source level. (1-8) |
| -Diverge to | Select a branching destination level. (1-8/END) |
| -Delete | Deletes the set items. |

■ **Displaying Latch**

Select [Latch Display] from the short-cut menu of the Sequence Window. The latch measurement result is displayed.

Figure 4.6-13 Latch Display Dialog



4.6.7 Stack

This section explains a SOFTUNE Workbench call stack.

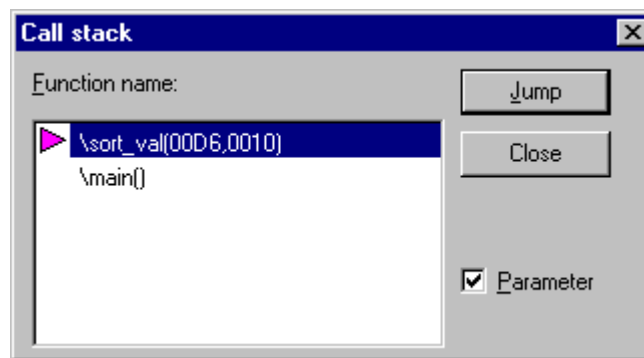
■ Call Stack

Usually, a program is a set of several subroutines. For this reason, as debugging advances, function calls of several stages occur. For example, one routine calls another and the called routine calls further another.

The call stack retains the relationship among function calls. Clicking a function name from the function name list or processing the jump button by clicking it immediately displays information for the function in the Source Window.

In that case, triangle symbol marks are displayed next to the function name.

Figure 4.6-14 Call Stack Dialog



The function written in the lowermost line of the function name list is the main function. This main function calls the function above it. The called function calls further a function above it. In this way, the function written in the uppermost line is the function in which the current PC exists.

When return is executed, functions are deleted in turn from the function name list, starting from the uppermost line.

-Parameter

When a check mark is set to the left of Parameter, an argument value is displayed after each function name, as shown in Figure 4.6-14.

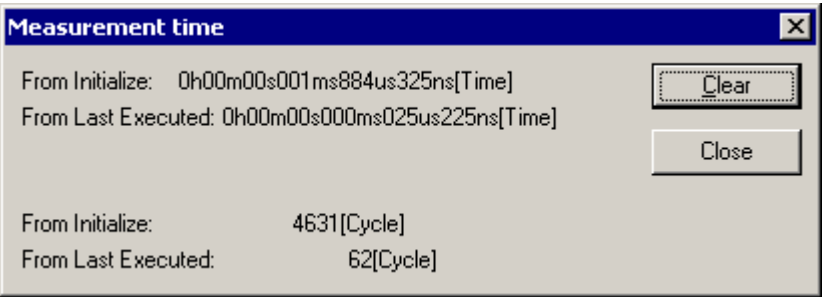
When no check mark is set to the left of Parameter, only parentheses "()" are displayed after each function name.

4.6.8 Time Measurement

This section explains SOFTUNE Workbench time measurement.

■ Time Measurement

Figure 4.6-15 Time Measurement Dialog



-Items to be displayed

	Time	Cycle count	Step count
SIM	×	○	○
EML(MB2141)	○	×	×
EML(MB2146-09)	×	×	×

-From Initialize

Indicates the cumulative total value of execution after the [Clear] button has been clicked.

-From Last Executed

Indicates the immediately preceding execution time.

Note:

The measurement results have errors. For details, refer to 'Measuring Execution Time' in each chapter of SOFTUNE Workbench User's Manual.

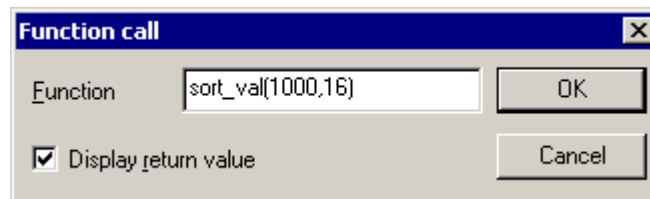
4.6.9 Call

This section explains the SOFTUNE Workbench function call function.

■ Function Call

The specified function can be started during debugging without reference to the flow of the program. This function is known as function call.

Figure 4.6-16 Function Call Dialog



When the function call dialog shown in Figure 4.6-16 opens, specify the function you want to call with a correct argument.

Compiles with C arguments, however, next argument does not specify.

- structure

- union

Figure 4.6-17 Function Call Result Dialog



When processing of the called function is terminated and control is returned, the function call result dialog shown in Figure 4.6-17 opens.

[Example]

When function definition is `int sub (int param);`, specify the function call as follows:

sub (10): When a constant value is directly specified

sub (ii): When variable ii is directly specified

■ Description

Execute the specified functions to display the return values. The return value is set to the built-in variable %CALL.

Evaluate the argument of the specified function in dummy argument format, and execute it.

If the count of specified actual arguments is more than the count of dummy arguments, evaluate as many counts of actual arguments as that exceeds the dummy arguments in an int type.

When the program hits a break point while the CALL command is executing a function, a break occurs at that position.

To execute the call command continuously, use the GO command.

To terminate the execution of the CALL command, use the CLEAR CALL command.

Note that it is not possible to nest this command.

The CALL command sets the break point for the current PC, and sets the return address so as to return to that point, calling a function.

Therefore, if the function called by the CALL command passes through the current PC, a break occurs in the middle of executing the function.

In such cases, the following message is displayed.

Break at address by Invalid call termination

Re-starts execution using the GO command to continue the execution of the CALL command.

Note:

The CALL function may change the resources such as the registers, memory or I/O from the state prior to the function call. To restore registers, hold contents prior to the function call and execute the functions, or use CLEAR CALL function.

Other resources are not restored.

4.6.10 Clear Call

This section explains the SOFTUNE Workbench call clear function.

■ Clear Call

"Clear Call" is used to restore the original state without executing the function call (see Section "4.6.9 Call") to the end. This function is used after program execution has been stopped by "Break Points...", etc.

When "Clear Call" is executed, control returns from the immediately called function. In this case, the function call result is not displayed because the called function is not executed to the end.

4.6.11 Vector

This section explains how to display and modify SOFTUNE Workbench vectors.

■ Vector

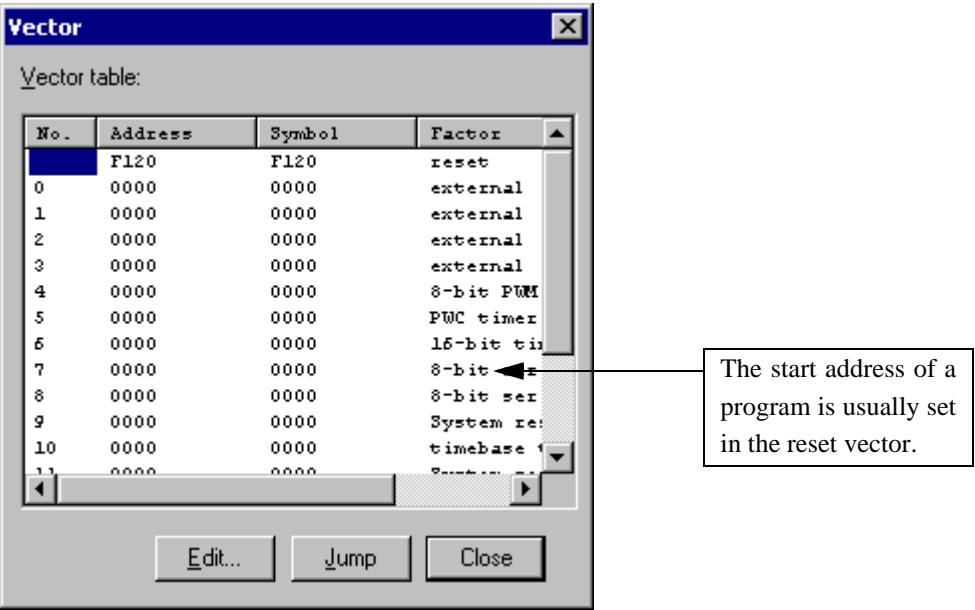
When the MCU is reset or when an interrupt processing request is issued for a variety of factors, the MCU sets the data, set in the address determined in advance according to the type of the interrupt, in the PC as the address of the interrupt processing routine. The address at which this interrupt processing routine is set is called a vector. Vectors are determined in advance according to the kind of the MCU.

■ Display and Setting Vectors

-Display

Figure 4.6-18 shows the vector display dialog.

Figure 4.6-18 Vector Display Dialog



Setting an address

Change the address, set in a vector as the following procedure:

- 1) Select a vector table number, then click the [Edit] button.
 - The vector edit dialog opens.
- 2) Set an address, then click the [OK] button.

■ Jump

Display the source of the stored program at the address set in the vector table in the following procedure:

- 1.Select a vector number.
- 2.Click the [Jump] button.

If the starting address of the program set in the vector table is incorrect, the source cannot be displayed (disassemble display).

Note:

The jump function merely displays the jump destination program; it does not update the program counter to move control to the address set in the vector table.

4.6.12 Load Target File

This section explains how to load the target file to be debugged by SOFTUNE Workbench.

■ Target File

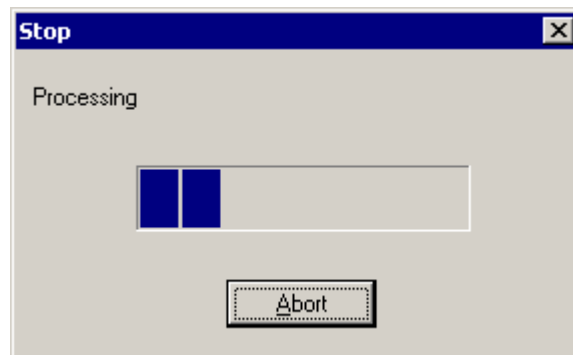
An ABS format target file is to be debugged. This file is registered as a project target file.

Debugging can be started after the ABS format target file has been created. Use SOFTUNE Workbench to create a source program and execute compile/assemble and link. Creation of the ABS format target file is enabled when the program is free from compile/assemble and link errors.

■ Loading the Target File

Before loading the target file, select [Start debug] from the [Debug] menu to place SOFTUNE Workbench in the debug session. When SOFTUNE Workbench enters the debug session, select [Load target file] from the [Debug] menu to load the target file. The target file load state display dialog shown in Figure 4.6-19 opens when the target file is being loaded.

Figure 4.6-19 Target File Load State Display Dialog



When loading the target file terminates, an entry point is set in the PC, the source line of the module including the entry point is displayed, and the program stops.

Execute [Step] and [Go], etc., subsequently to continue debugging.

4.6.13 Start Debug/End Debug

This section explains how to start and end debug session.

■ Start Debug

"Start debug" places SOFTUNE Workbench in the debug session to enable the subsequent use of debugger commands. When SOFTUNE Workbench enters the debug session, at first load the target file (see Section "4.6.12 Load Target File").

■ End Debug

"End debug" terminates the SOFTUNE Workbench debug session.

4.7 Setup

"Setup" sets SOFTUNE Workbench execution environment, debugger mode environment, and other tools.

■ Setting the SOFTUNE Workbench Execution Environment

- Development

■ Setting the Debugger Mode Environment

- Debug Environment
- Memory Map...
- Flash area control

■ Setting Other Tools

- Tool
- Keyboard
- Editor
- Error
- Tool Startup

4.7.1 Development

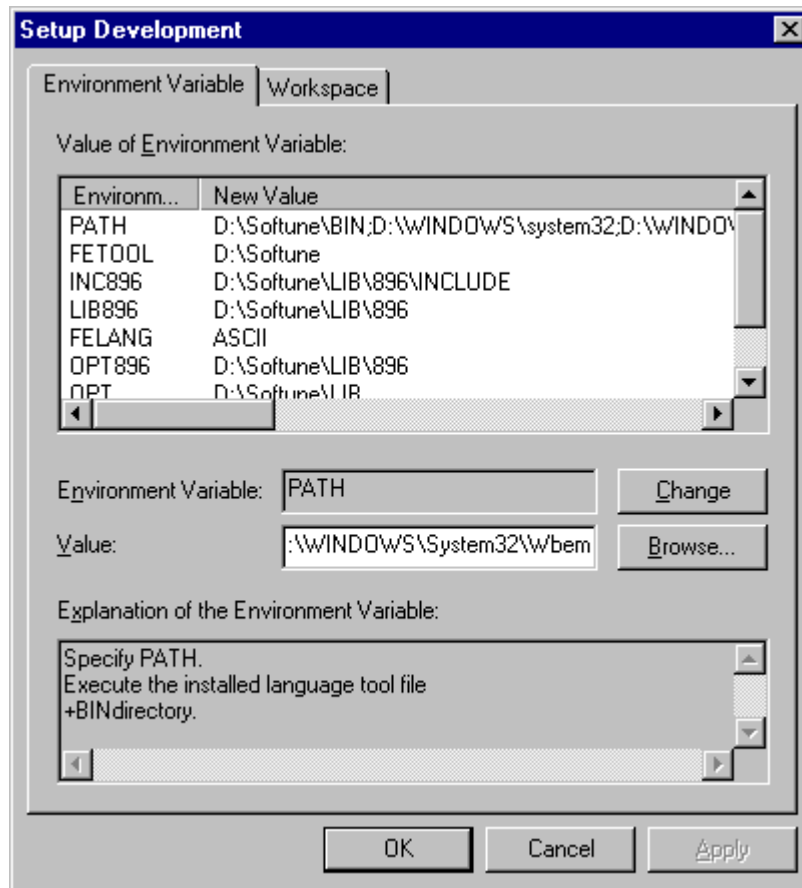
"Development..." sets SOFTUNE Workbench operation and the environment variables required by language tools (e.g., compiler).

■ Environment Variable

Figure 4.7-1 shows the environment variable setup dialog. The environment variables listed in Table 4.7-1 are set in this section.

Table 4.7-1 Environment Variable Names

Environment variable name	Explanation
FETOOL	Standard directory in which language tools were installed
INC896	Directory in which include file exists
LIB896	Directory in which library file exists
OPT896	Directory in which the language tool default option file exists
FELANG	Character code system switching in the messages output by language tools (SJIS: Japanese language (shift JIS), ASCII: English)
PATH	Directory in which language tools exist
TMP	Directory in which work files exist

Figure 4.7-1 Development Environment Setup (Environment Variable) Dialog

Set the development environment in the following procedure:

1. Select the name of the environment variable whose setting is to be changed from the [Value of Environment Variable] list.
The current setting value is displayed in the [Value] field.
Simple explanation of the environment variable is displayed in [Explanation of the Environment Variable].
2. Change the description of the [Value] field.
3. Click the [Change] button.

■ Workspace

"Workspace" sets the following SOFTUNE Workbench operations:

-Open the last workspace at starting

Setting a check mark to the left of this item enables the opening of the previously opened workspace file when SOFTUNE Workbench is started.

-Output tool option at compile/assemble

Setting a check mark to the left of this item enables the display of the options, specified when the C compiler or assembler is started, in the Output Window.

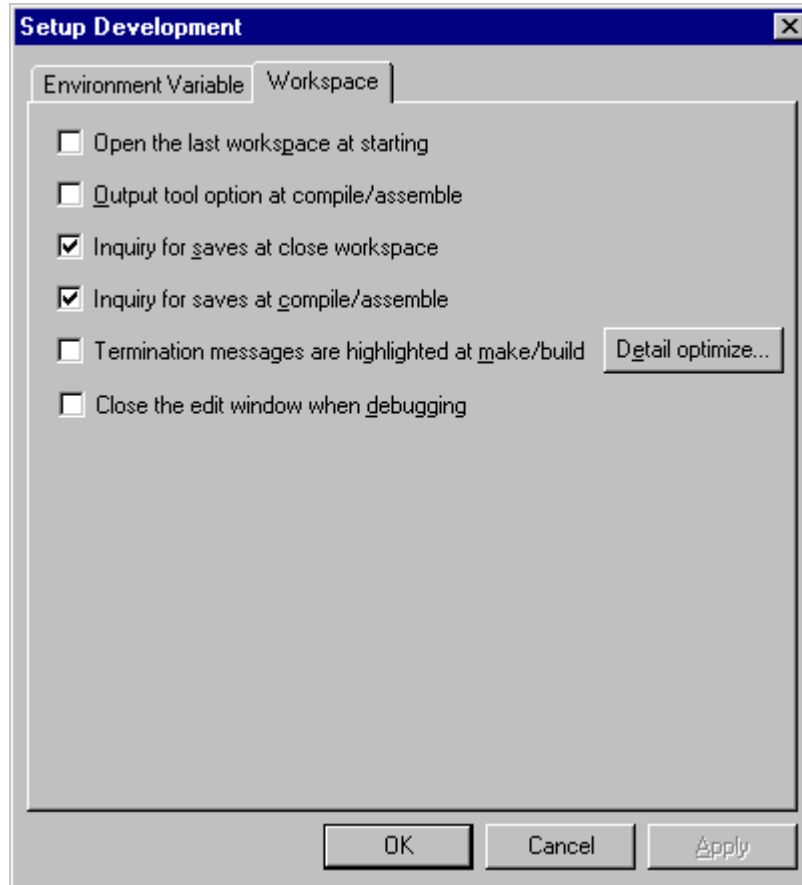
-Inquiry for save at close workspace

When a check mark is set to the left of this item, SOFTUNE Workbench asks you whether to save the currently open file to the workspace file when the workspace is closed.

-Inquiry for save at compile/assemble

When a check mark is set to the left of this item, SOFTUNE Workbench asks you whether to save the file currently being edited before compile/assemble.

Figure 4.7-2 Development Environment Setup (Workspace) Dialog

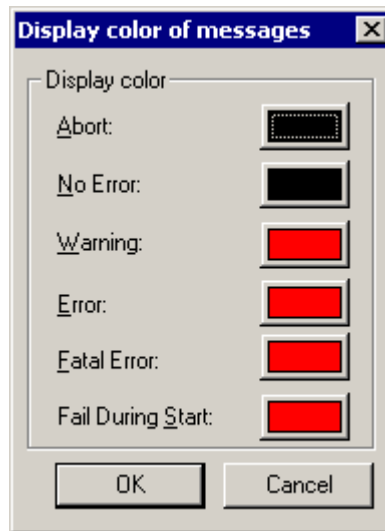


-Termination messages are highlighted at make/build

When this item is checked, the display color of termination messages (abort, no error, warning, error, fatal error, or fail during start) during Compile, Assemble, Make, or Build, can be changed.

To change the display color, click the [Detailed optimize] button at the right of this field; the termination message display color dialog is displayed (Figure 4.7-3). Change the display color.

Figure 4.7-3 Display Color of Messages Dialog



- Close the edit window when debugging
By checking this item on a box, edit window, which opens as debugger is started, will be closed.

4.7.2 Debug Environment

"Debug Environment" sets the debug environment; it is valid only when SOFTUNE Workbench is in the debug session.

■ Items to be Set

Select and set the following items from the submenu:

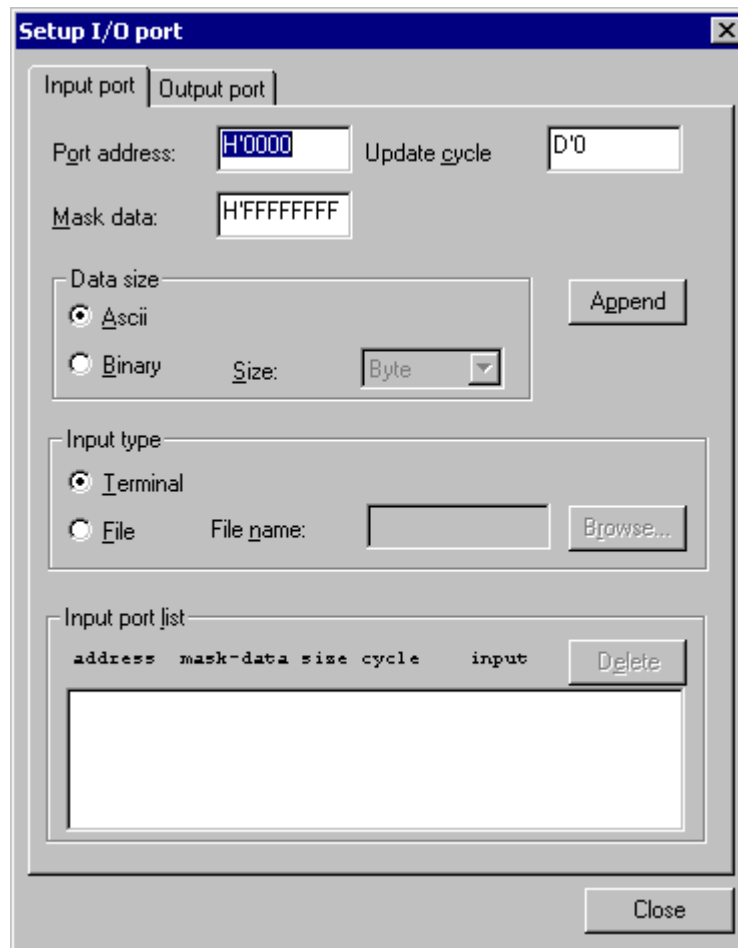
- I/O Port
- Interrupt
- Debug Environment
- Setup Wizard

4.7.2.1 I/O Port

This section explains the I/O port setup procedure.

■ Setting an Input Port

Figure 4.7-4 Input Port Setup Dialog



-Port Address

Specifies a port address.

-Mask Data

Specifies address mask. Only the addresses whose bits are 1 are to be compared.

-Data Size

Specifies a data input type. When Binary is selected, specify its size. (Byte/Word/Long)

-Input Type

Specifies a port data input source.

-Input Port List

Displays the currently specified ports.

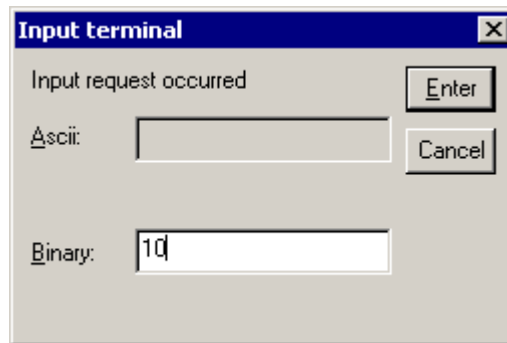
-Input terminal

When an input request is issued during program execution with [Input Type] set to [Terminal] in input port setting, the input terminal dialog opens.

Specifying [ASCII] as [Data Type] in input port setting enables ASCII input.

Specifying [Binary] as [Data Type] enables binary input.

Figure 4.7-5 Input Terminal Dialog

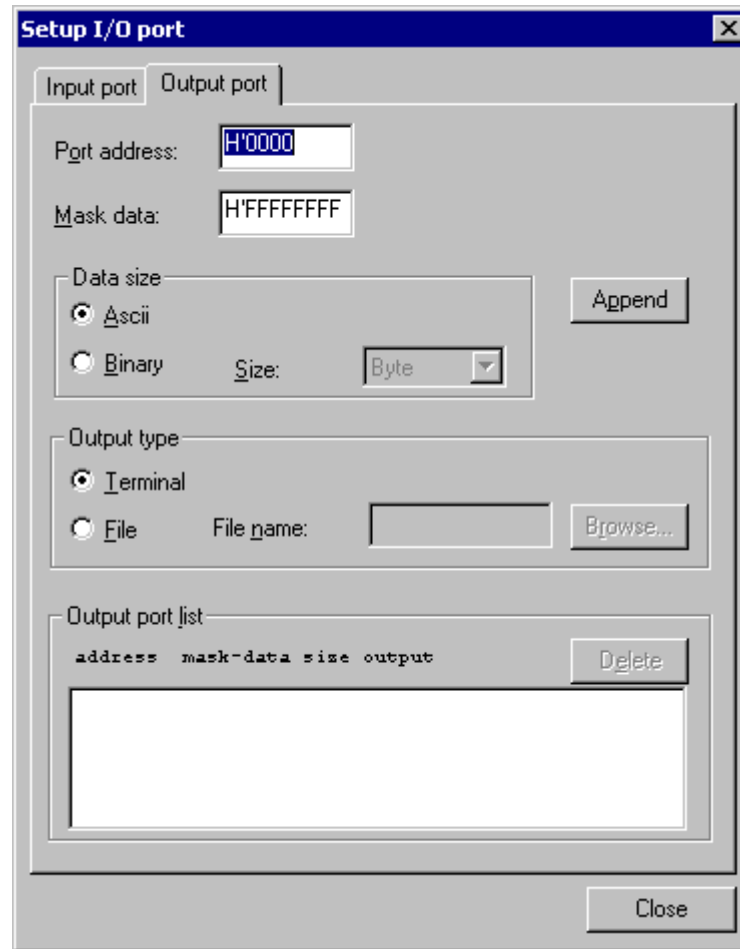


■ Resetting an Input Port

1. Select the input ports to be reset from [Input Port List].
2. Click the [Delete] button.
3. When resetting all the selected ports is completed, click the [Close] button.

■ Setting an Output Port

Figure 4.7-6 Output Port Setup Dialog



-Port Address

Specifies a port address.

-Mask Data

Specifies address mask. Only the addresses whose bits are 1 are to be compared.

-Data Size

Specifies a data output type. When Binary is selected, specify its size. (Byte/Word/Long)

-Output Type

Specifies a port data output destination.

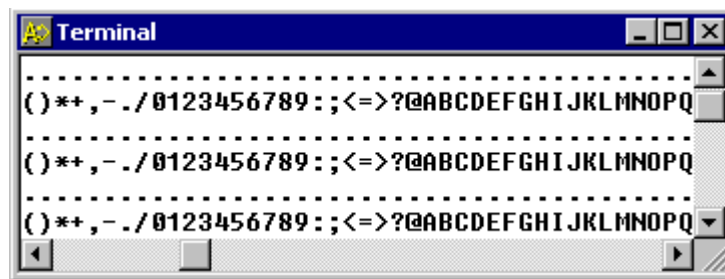
-Output Port List

Displays the currently specified ports.

-Output terminal

When an output request is issued during program execution with [Output Type] set to [Terminal] in output port setting, the Terminal Window is displayed. The output type also depends on [Data Type].

Figure 4.7-7 Output Terminal Window (Binary)



■ Resetting an Output Port

1. Select the output ports you want to reset from [Output Port List].
2. Click the [Delete] button.
3. When resetting all the selected ports is completed, click the [Close] button.

4.7.2.2 Interrupt

This section explains the interrupt setup procedure.

■ Setting an Interrupt

1. When the interrupt setup dialog shown in Figure 4.7-8 opens, set an [Interrupt number].
2. Select an [Request timing].
[One Time] or [Interval] can be selected.
3. Set the [Cycle count].
4. Click the [Append] button.
The set interrupt number, issuance timing, and interrupt cycle count are displayed in [Interrupt List].
5. When setting all the items is completed, click the [Close] button.

■ Resetting an Interrupt

1. When the interrupt setup dialog shown in Figure 4.7-8 opens, set an [Interrupt number].
2. Select the interrupt to be reset from [Interrupt List].
3. Click the [Delete] button.
4. When resetting all the selected interrupts is completed, click the [Close] button.

Figure 4.7-8 Interrupt Setup Dialog

Number	Request timing	cycle
11	One time	1000

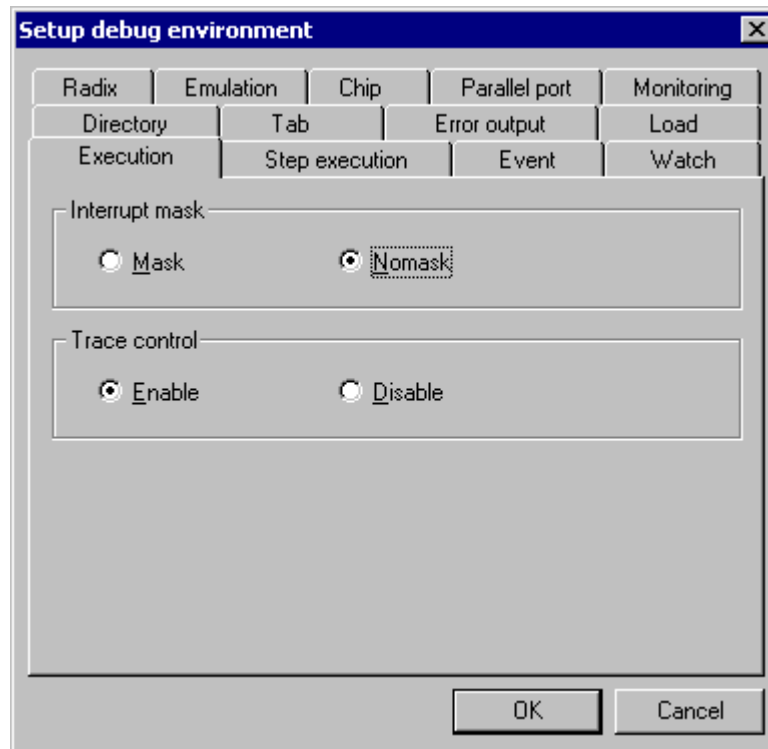
4.7.2.3 Debug Environment

This section explains the debug environment setup procedure.

■ Debug Environment Setup Procedure

-Execution (MB2141)

Figure 4.7-9 Debug Environment Setup Dialog (Execution)



-Interrupt Mask

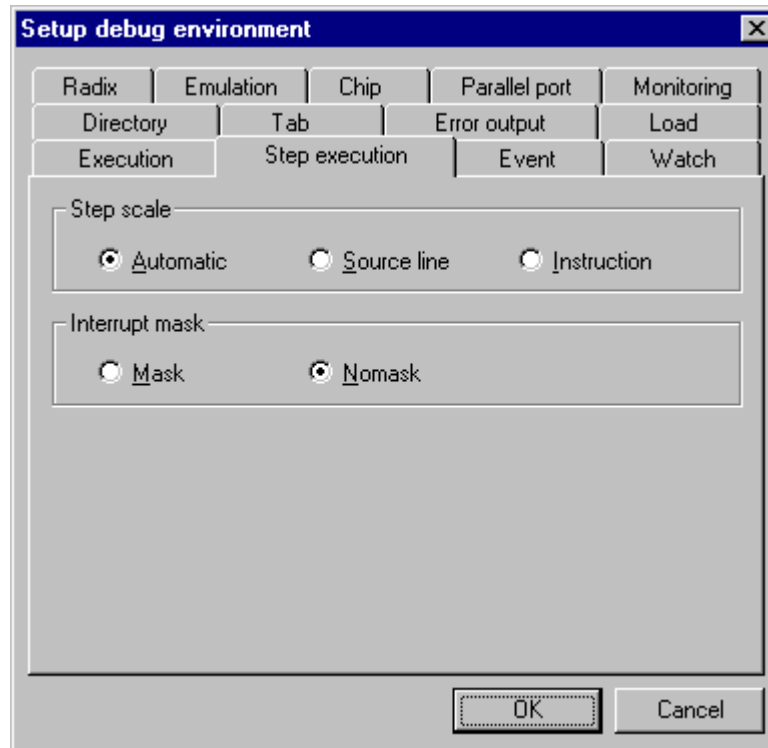
Specifies whether to enable/disable interrupt mask during step execution.

-Trace Control

Specifies whether to enable/disable trace control.

-Step Execution [MB2141/MB2146-09]

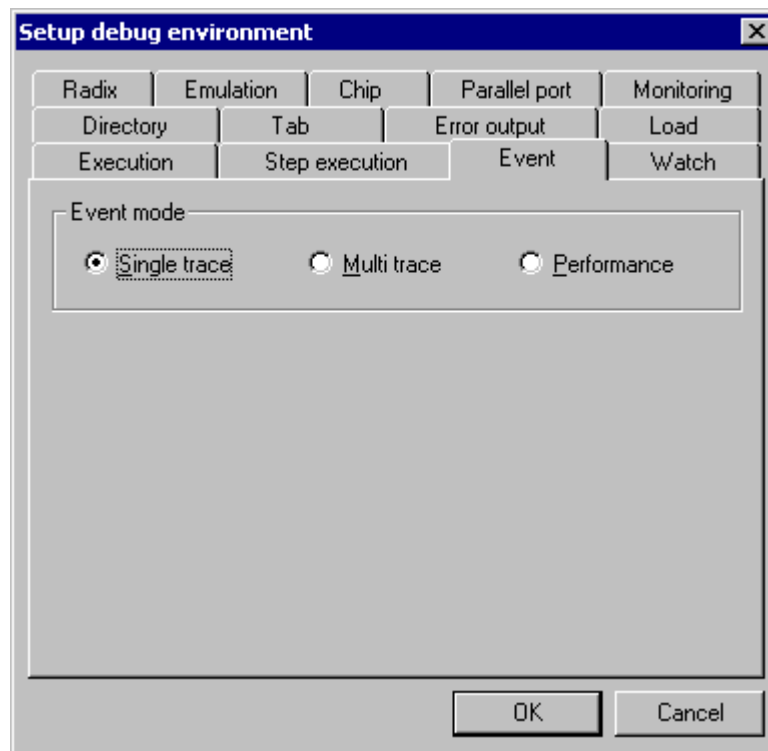
Figure 4.7-10 Debug Environment Setup Dialog (Step Execution)



- Step Scale
 - Automatic
Automatically sets the step unit according to the window display state.
 - Source Line
Executes the step in units of source lines.
 - Instruction
Executes the step in units of machine languages.
- Interrupt Mask
Specifies whether to enable/disable interrupts.

-Event [MB2141]

Figure 4.7-11 Debug Environment Setup Dialog (Event)



-Event Mode

Specifies an event mode.

-Single Trace

Specifies the normal mode.

-Multi-Trace

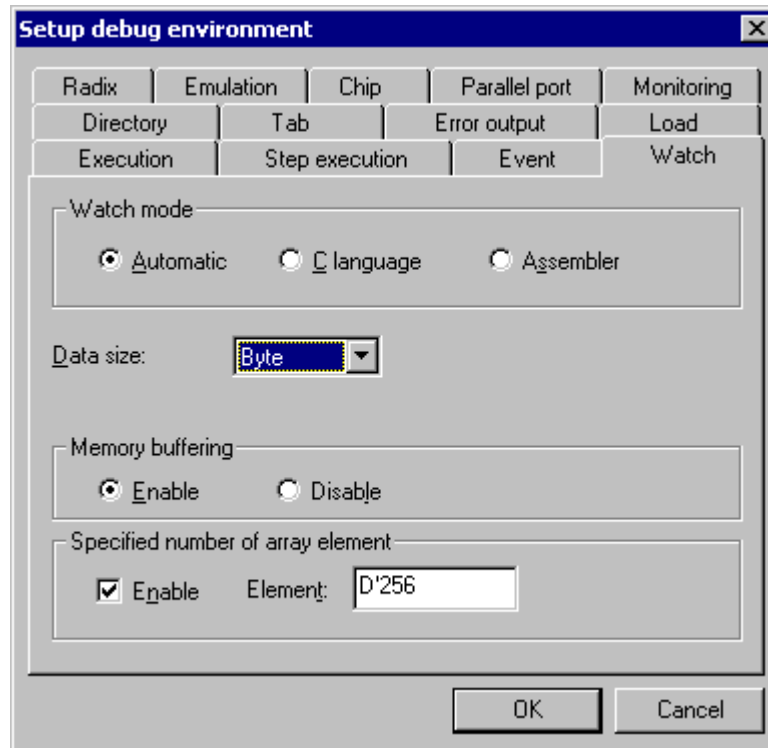
Specifies the multi-trace mode.

-Performance

Specifies the performance mode.

-Watch

Figure 4.7-12 Debug Environment Setup Dialog (Watch)



-Watch Mode

-Automatic

Sets the watch mode automatically according to the analysis result.

-C Language

Sets the C language mode (interpretation as C language expressions).

-Assembler

Sets the assembler mode (interpretation as assembler expressions).

-Data Size

Sets the display size in the assembler mode.

-Byte/word/long/single precision/double precision

-Memory Buffering

-Enable

In case of variables as arrays or structures, the memory of whole variables is read.

They are accessed by size of the top variable.

-Disable

The default of this control is "Enable".

If "Disable" is selected, the watch window and the local window may be displayed slowly.

In case of variables as arrays or structures, the memory of each element and member unit is read.

-Specified number of array element

-Enable

Debugger displays a warning dialog in case of array element larger than the number of limited array-element when you registering or expanding an array with a watch variable.

-Element

Specifies number (a default is D'256) of array element.

The default of this control is "Enable".

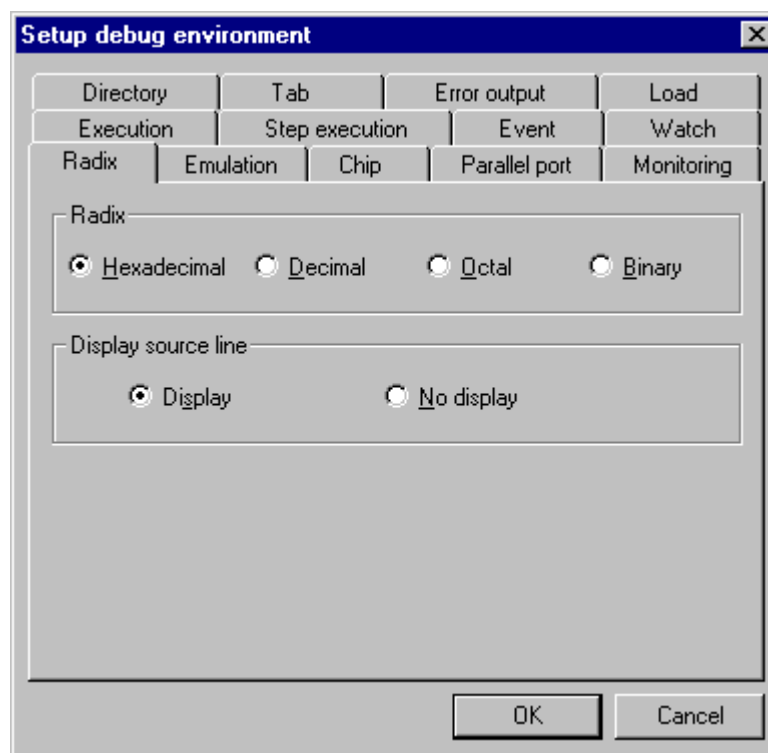
If "Disable" is selected, the watch window and the local window may be displayed slowly.

Note:

If memory buffering is set as valid, correct value such as I/O to request read of fixed size cannot be displayed.

-Radix

Figure 4.7-13 Debug Environment Setup Dialog (Radix)



-Radix

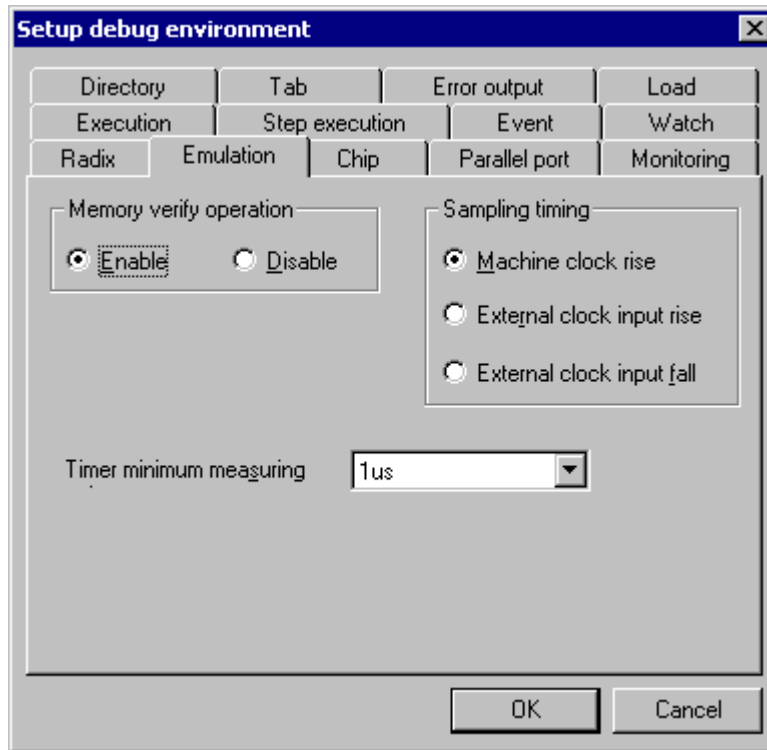
Sets the radix for numerical value display and analysis.

-Display Source Line

Switches source line display and nondisplay.

-Emulation [MB2141]

Figure 4.7-14 Debug Environment Setup Dialog (Emulation)



-Memory Verify Operation

Specifies whether to verify memory when data is written to memory.

-Sampling Timing

- Machine Clock rise

Sets the rising of the machine clock as sampling timing.

- External Clock Input rise

Sets the rising of the external clock as sampling timing.

- External Clock Input fall

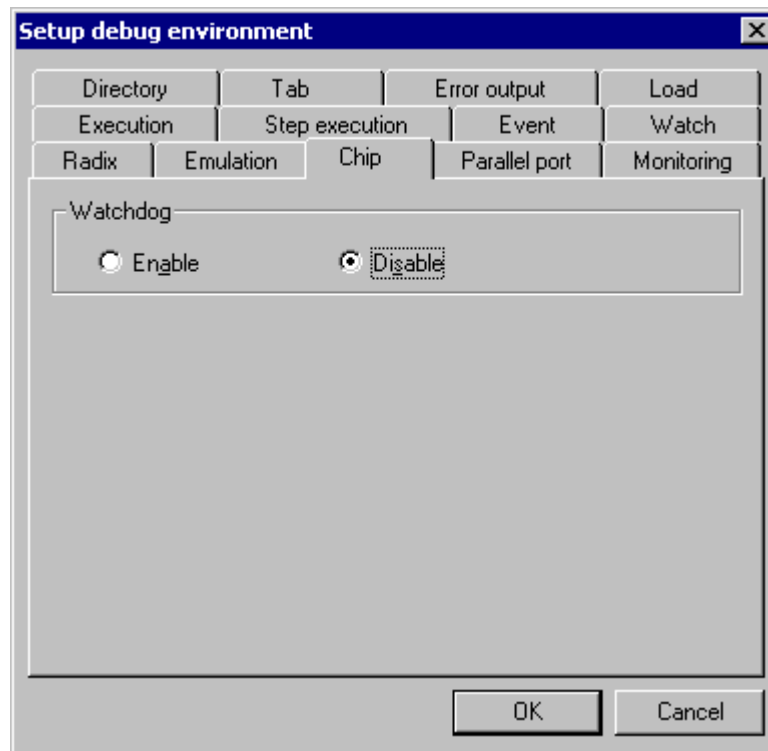
Sets the falling of the external clock as sampling timing.

-Timer minimum measuring unit

Specifies the minimum unit for execution time measurement. (1 μ s/100ms)

- Chip [MB2141/MB2146-09]

Figure 4.7-15 Debug Environment Setup Dialog (Chip)

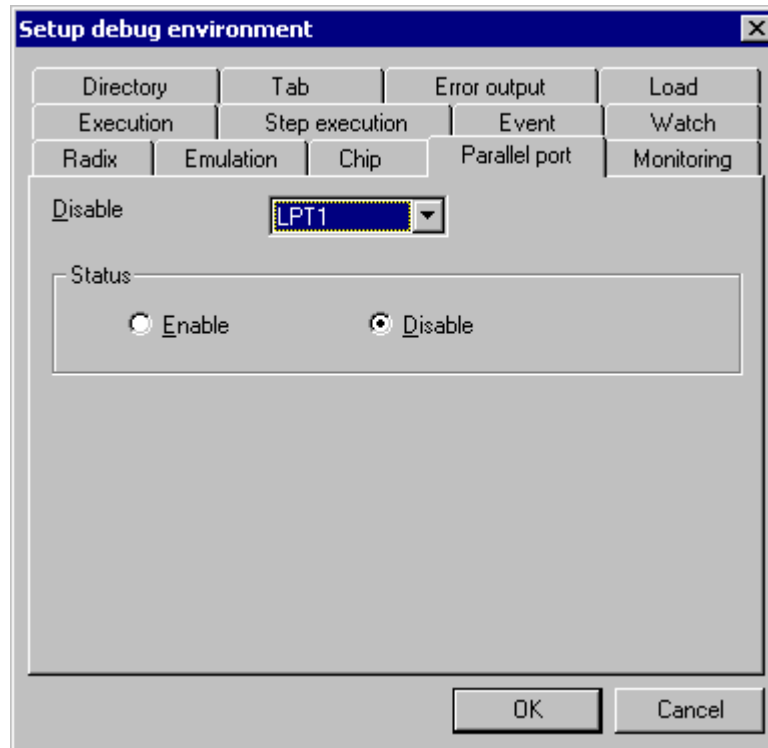


- Watchdog

Specifies whether the watchdog timer is valid or invalid.

-Parallel Port [MB2141]

Figure 4.7-16 Debug Environment Setup Dialog (Parallel Port)



-Disable

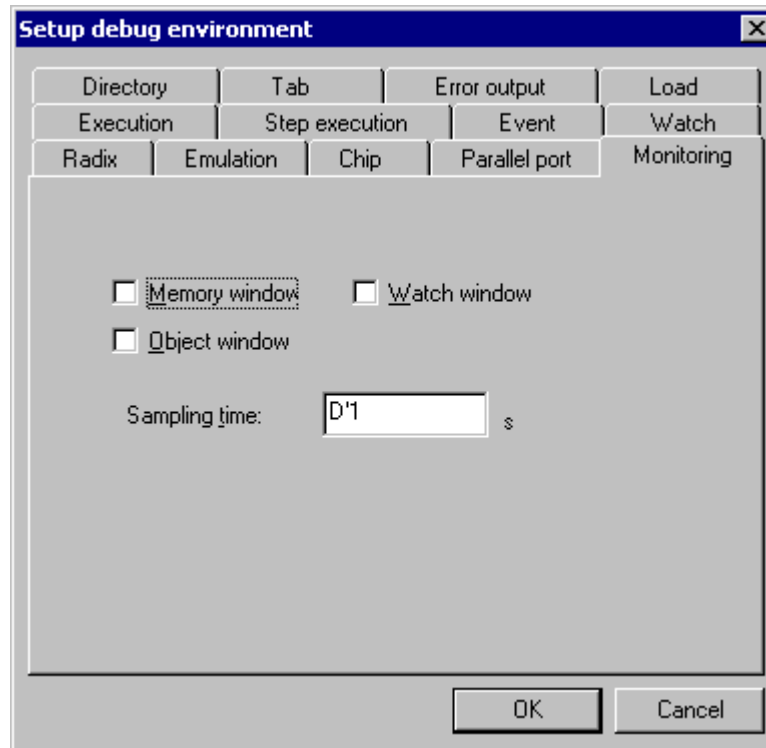
Specifies non-connection or a parallel port name.(None/LPT1/LPT2)

-Status

Specifies whether the parallel port is valid or invalid.

-Monitoring[MB2141/Simulator]

Figure 4.7-17 Debug Environment Setup Dialog (Monitoring)



-Monitoring Control

Sets monitoring control.

-Memory Window

Specifies whether to monitor the Memory Window.

-Watch Window

Specifies whether to monitor the Watch Window.

-Object Window

Specifies whether to monitor the Object Window.

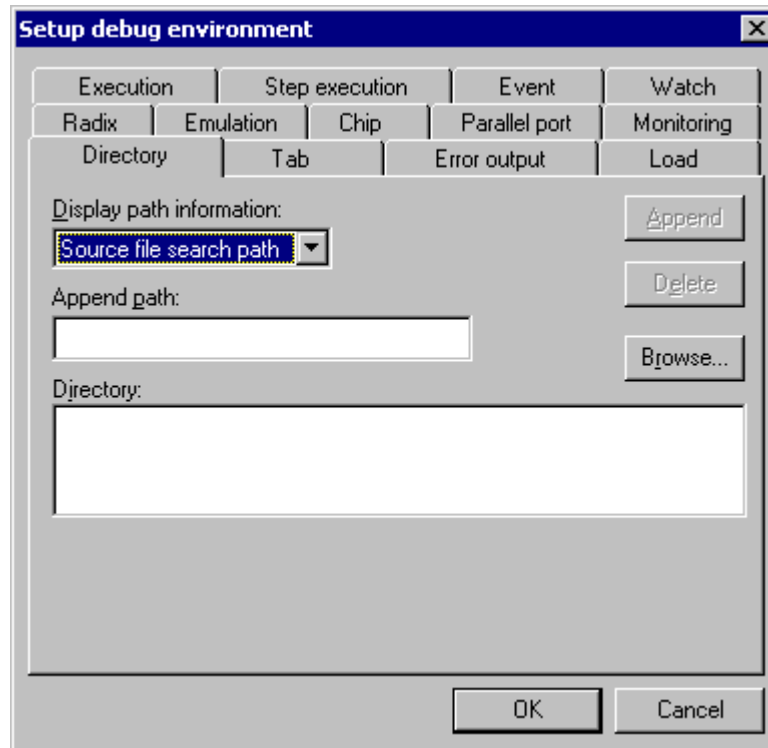
-Sampling Time

Specifies sampling time.

MB2141 : minimum 1000ms

-Directory

Figure 4.7-18 Debug Environment Setup Dialog (Directory)



-Display Path Information

Specifies the path information to be displayed.

-Append Path

Sets the path to be added.

-Directory

Displays the currently set directory.

-Directory setup procedure

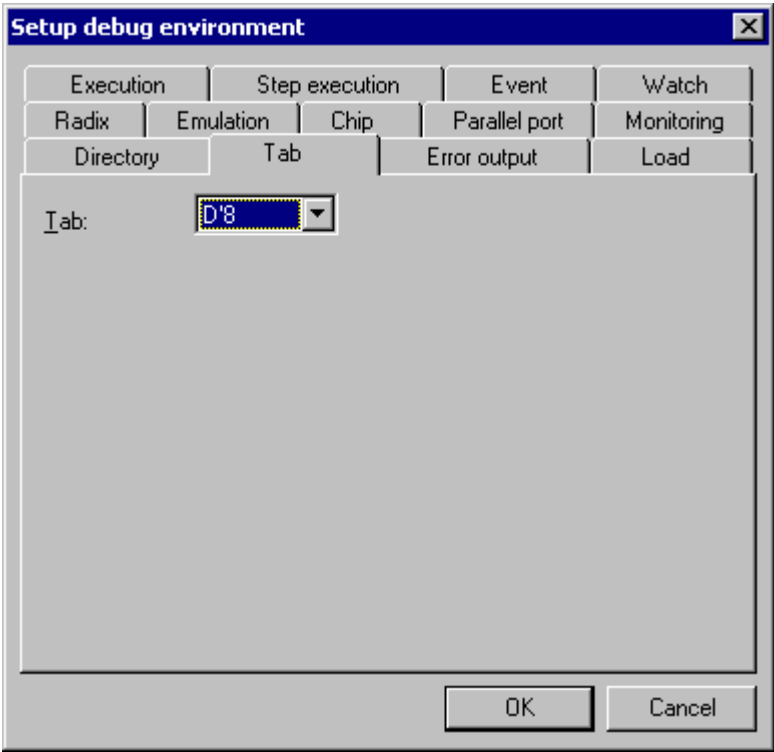
1. Select the [Display Path Information] to be displayed.
2. Set the [Append Path].
Clicking the [Browse] button to the right of the [Append Path] setup field enables path selection.
3. Click the [Append] button.
4. When there is no other item to be set, click the [OK] button.

-Directory reset procedure

1. Select the directory you want to delete from [Directory].
2. Click the [Delete] button.
3. When there is no other item to be set, click the [OK] button.

-Tab

Figure 4.7-19 Debug Environment Setup Dialog (Tab)

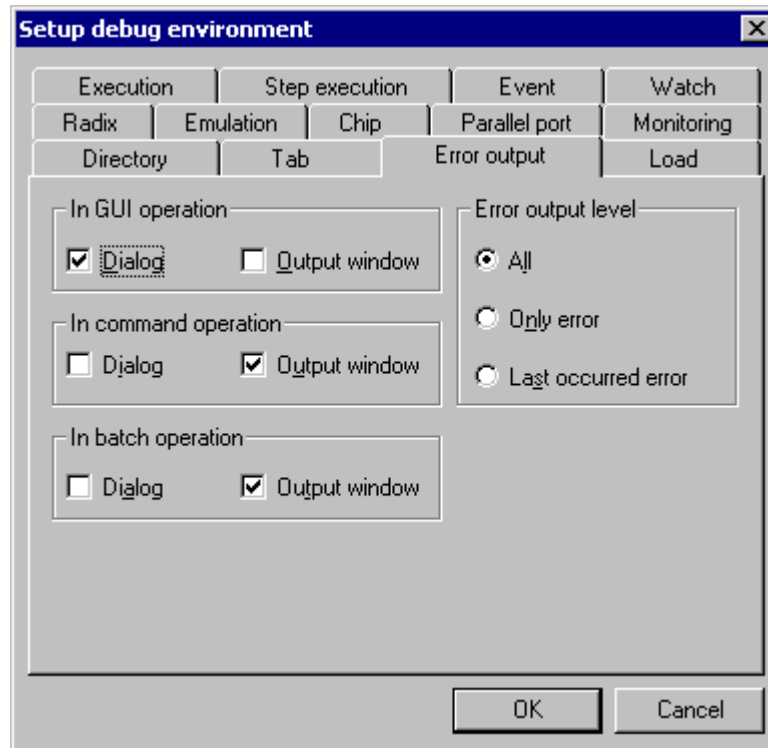


-Tab

Specifies the Tab.(D'4/D'8)

-Error output

Figure 4.7-20 Debug Environment Setup Dialog (Error Output)



-In GUI Operation

Specifies an error output type at GUI operation.

-In Command Operation

Specifies an error output type at command operation.

-In Batch Operation

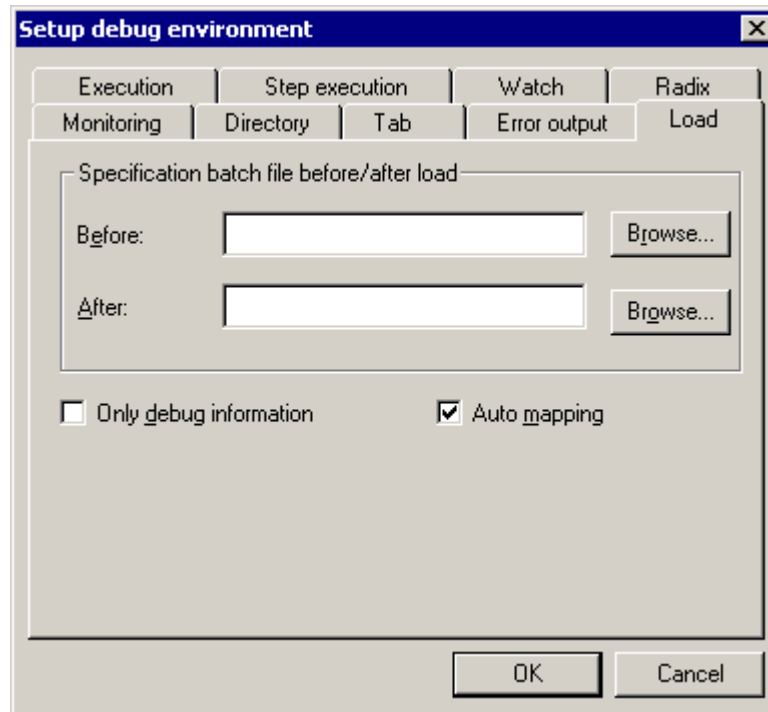
Specifies an error output type at batch operation.

-Error Output Level

Sets the output type when several errors occur.

-Load

Figure 4.7-21 Debug Environment Setup Dialog (Load)



This sets the environment when loading a target file registered in the project.

- Specification batch file before/after load

Before

This specifies the batch file to execute prior to the loading of the target file. This can be changed using the Debugger's setup wizard.

After

This specifies the batch file to execute after the loading of the target file. This can be changed using the debugger's setup wizard.

- Only debug information

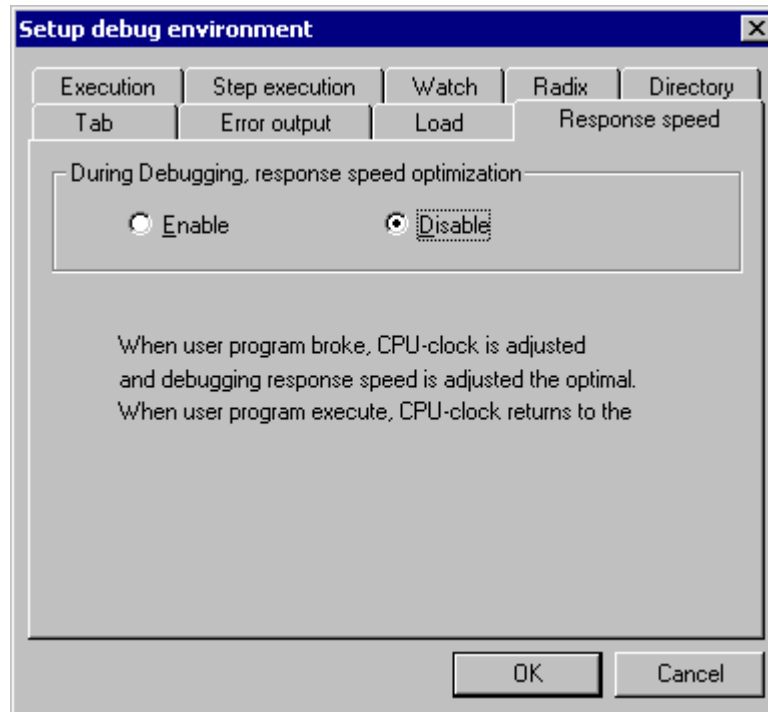
This specifies whether or not to load only the debug information or not. When checked, only the debug information is loaded.

- Auto mapping

This specifies whether or not to enable the Auto-Map Setting. When checked, Auto-Map Setting is enabled. This is valid the simulator only.

- Response speed [MB2146-09]

Figure 4.7-22 Debug Environment Setup Dialog (Response Speed)



- During Debugging, response speed optimization

When the user program is broken, the CPU clock is automatically adjusted, and it specifies whether to set the response speed of the debugger to the optimal value. Further, when the user program is executed, be sure to return the CPU clock to the original value.

4.7.2.4 Setup Wizard

This section explains how to operate the debugger's Setup Wizard.

■ Setup Wizard Operation Procedure

1. Select [Setup Name] from [Debugger Startup Selection], then click the [OK] button.
The check dialog opens.
2. Click the [OK] button.
Setup Wizard is started.
3. Click the [Next] button.
4. Select a debugger type, then click the [Next] button.
Setting of the subsequent items depends on the debugger's type.
To reset an item, click the [Return] button.
The immediately preceding setup screen is redisplayed.

■ Procedure When the Emulator (MB2141) is Selected

1. Select the emulator type.
2. Click the [Next] button.
3. Select [RS232C] or [LAN] as the device type.
When [RS232C] is selected, set a port and baud rate.
When [LAN] is selected, set a host name.
4. Click the [Next] button.
5. Set whether or not to automatically load the monitor program at debugging start.
6. Click the [Next] button.
7. Set whether or not to load the target file automatically during debugging.
8. Specify the batch files used before and after load as required.
Clicking the [Refer] button to the right of each input field enables batch file selection.
9. Click the [Next] button.
10. Select the item to be set in the setup file.
When [Specify] is selected, click the [Setup] button to open the item selection dialog, select the item to be set from the dialog, then click the [OK] button.
11. Click the [Next] button.
12. Click the [Complete] button.
Emulator (MB2141) setup has been just completed.

■ Procedure When the Emulator (MB2146-09) is Selected

1. Select the emulator type.
2. Click the [Next] button.
3. Select ([USB]) as the device type.
4. Click the [Next] button.
5. Specify the oscillation frequency to be useful (F_{CH}) and clock up mode.
6. Click the [Next] button.
7. Set whether or not to load the target file automatically during debugging.
8. Specify the batch files used before and after load as required.
Clicking the [Refer] button to the right of each input field enables batch file selection.
9. Click the [Next] button.
10. Select the item to be set in the setup file.
When [Specify] is selected, click the [Setup] button to open the item selection dialog, select the item to be set from the dialog, then click the [OK] button.
11. Click the [Next] button.
12. Click the [Complete] button.
Emulator (MB2146-09) setup has been just completed.

■ Procedure when the Simulator Debugger is Selected

1. Set whether to load the target file automatically during debugging.
2. Specify the batch files used before and after load as required.
Clicking the [Refer] button to the right of each input field enables batch file selection.
3. Click the [Next] button.
4. Select the item to be set in the setup file.
When [Specify] is selected, click the [Setup] button to open the item selection dialog, and select the item to be set from the dialog, then click the [OK] button.
5. Click the [Next] button.
6. Click the [Complete] button.
Simulator debugger setup has been just completed.

■ Procedure when the monitor debugger is selected

1. Enter a password required at starting debugger.
2. Click the [Next] button.
3. Select [USB] as the device type.
4. Input the oscillation frequency.
5. Set whether or not to load the target file automatically during debugging.
6. Specify the batch files used before and after load as required.
Clicking the [Refer] button to the right of each input field enables batch file selection.
7. Click the [Next] button.
8. Select the item to be set in the setup file.
When [Specify] is selected, click the [Setup] button to open the item selection dialog, and select the item to be set from the dialog, then click the [OK] button.
9. Click the [Next] button.
10. Click the [Complete] button.
Monitor debugger setup has been just completed.

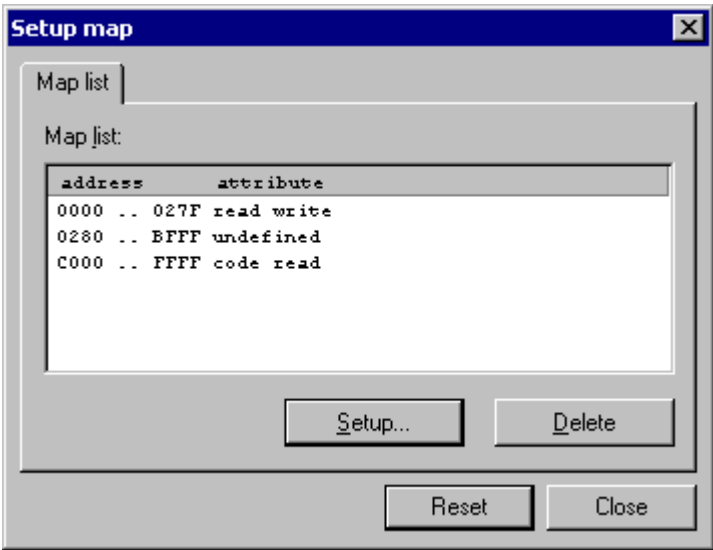
4.7.3 Memory Map

"Memory Map..." sets the debugger's memory map.

■ Memory Map Setup [Simulator Debugger]

-Map List

Figure 4.7-23 Memory Map Setup Dialog



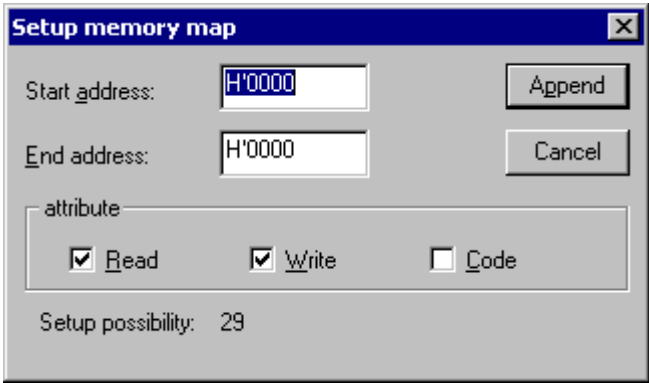
-Map list

Displays the currently set map area.

-Clicking the [Setup] button opens the setup dialog (Figure 4.7-23) corresponding to the debug session.

-Map Setup

Figure 4.7-24 Map Setup Dialog

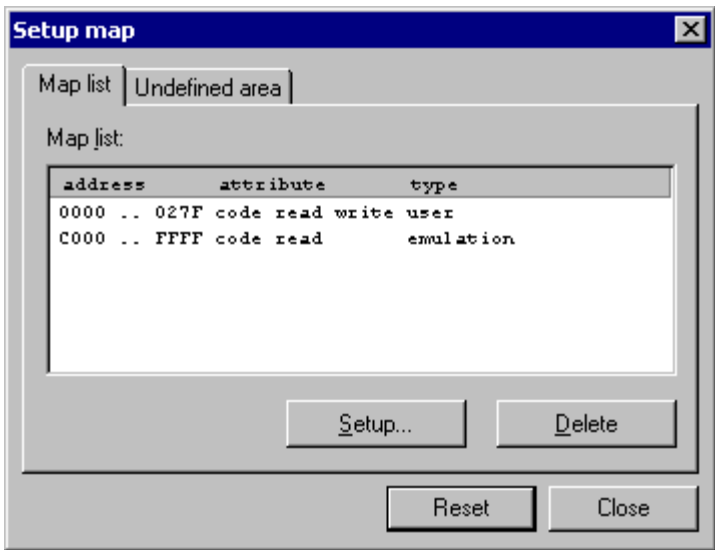


- Start Address
Specifies the start address to be set.
- End Address
Specifies the end address to be set.
- Attribute
Specifies a memory space attribute (Read, Write, or Code).

■ Memory Map Setup [MB2141 (Emulator)]

-Map List

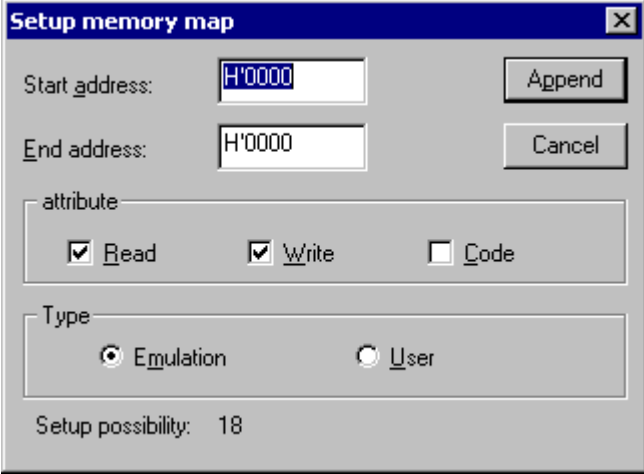
Figure 4.7-25 Memory Map Setup Dialog



- Map list
Displays the currently set map area.
- Clicking the [Setup] button opens the setup dialog (Figure 4.7-25) corresponding to the debug session.

-Map Setup

Figure 4.7-26 Map Setup Dialog



The dialog box is titled "Setup memory map" and has a close button (X) in the top right corner. It contains the following fields and controls:

- Start address:** A text box containing "H'0000" with an "Append" button to its right.
- End address:** A text box containing "H'0000" with a "Cancel" button to its right.
- attribute:** A group box containing three checkboxes: ☒ Read, ☒ Write, and ☐ Code.
- Type:** A group box containing two radio buttons: ☒ Emulation and ☐ User.
- Setup possibility:** A label followed by the number "18".

-Start Address

Specifies the start address to be set.

-End Address

Specifies the end address to be set.

-Attribute

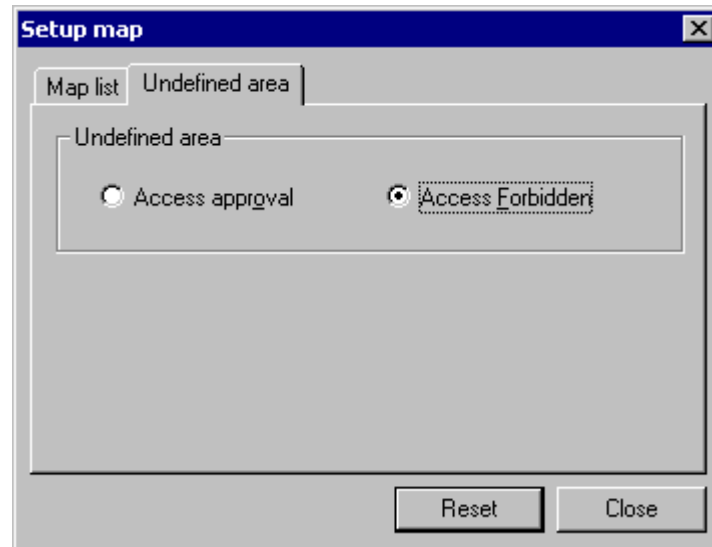
Specifies a memory space attribute (Read, Write, or Code).

-Type

Specifies a setup area type.

-Undefined Area [MB2141(emulator)]

Figure 4.7-27 Undefined Area Setup Dialog



- Undefined Area
 - Access Approval
Permits access to the undefined area.
 - Access Forbidden
Inhibits access to the undefined area.

4.7.4 FLASH Area Control

The MB2146-09 emulator supports programming to the Flash memory area.

■ FLASH Area Control

The MB2146-09 emulator supports functions of programming to the Flash memory area and of code breaks (software breaks). The emulator saves the contents of the Flash memory area in the debugger's buffer; and references the buffer's contents when reading from/programming to the Flash memory. Writing to Flash memory is usually performed automatically prior to executive operation or reset processing.

■ Download FLASH Memory

([Environment]-[FLASH area control]-[Download FLASH memory] menu)

Updates Flash memory. Flash memory is usually updated automatically prior to executive operation or reset processing. Use this menu when updating Flash memory before automatic updating. This menu is enabled when data in the Flash memory area is changed, requiring the writing to of Flash memory.

■ Upload FLASH Memory

([Environment]-[FLASH area control]-[Upload FLASH memory] menu)

Synchronizes Flash memory and the buffers within the Debugger. Be sure to perform this synchronization when Flash memory is rewritten (updated) by the user program, or the program would not operate properly.

■ Erase FLASH Memory

([Environment]-[FLASH area control]-[Erase FLASH memory] menu)

Erase all data in Flash memory. Note that this operation will erase all code break (software break) settings.

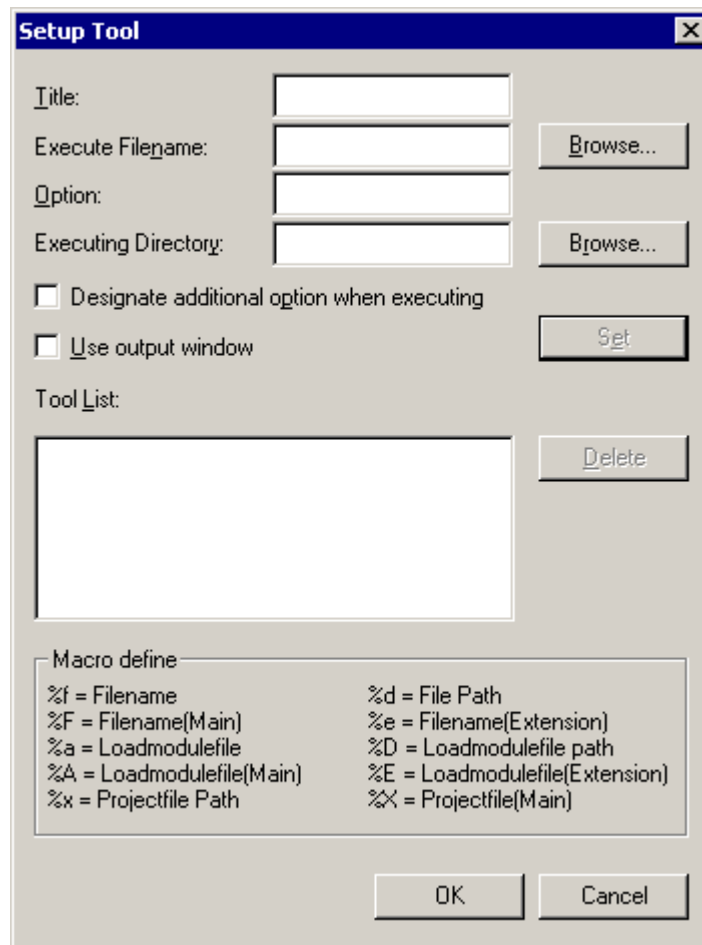
4.7.5 Tool

"Tool..." sets the tools to be directly started by SOFTUNE Workbench.

■ Tools

"Tool..." is not a tool that takes charge of basic SOFTUNE Workbench functions such as a C compiler and assembler. It is a function that builds auxiliary tools (e.g., simple filters) into the system so that they can be started directly from SOFTUNE Workbench. Building "dir" into the system, for example, enables the output of the result obtained as a result of executing the dir command at the DOS prompt to the SOFTUNE Workbench Output Window.

Figure 4.7-28 Tool Setup Dialog



■ Tool Setup Procedure

1. Select [Tool...] from the [Setup] menu.
The tool setup dialog shown in Figure 4.7-28 opens.
2. Set a title that differs from the registered names.
3. Specify the execution file name of the tool to be registered.
Clicking the [Browse] button to the right of this field opens the file selection dialog. The execution file

name of the tool can be selected from this dialog.

4. Set an option.

Macro description can be used in this field. For macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" in SOFTUNE Workbench User's Manual.

5. Write an executing directory.

This description may be omitted if control need not be moved to any specific executing directory.

Clicking the [Browse] button to the right of this field opens the file selection dialog. A run-time directory can be selected from this dialog.

6. Set a check mark to the left of [Designate Additional Option when Executing] and [Use Output Window] as required.

When a check mark is set to the left of [Designate Additional Option when Executing], SOFTUNE Workbench asks you to enter additional options when a tool is started. When a check mark is set to the left of [Use Output Window], SOFTUNE Workbench displays tool output (output to the standard output device or standard error output device) in the Output Window.

7. Click the [Setup] button.

■ Tool Deletion Procedure

1. Select [Tool...] from the [Setup] menu.

The tool setup dialog shown in Figure 4.7-28 opens.

2. Select the tool title you want to delete from the tool list.

3. Click the [Delete] button.

■ Tool Change Procedure

1. Select [Tool...] from the [Setup] menu.

The tool setup dialog shown in Figure 4.7-28 opens.

2. Select the tool title you want to change from the tool list.

The values set in [Title], [Execute Filename], [Option], [Executing Directory], [Designate Additional Option when Executing], and [Use Output Window] are displayed.

3. Change [Execute Filename], [Option], [Executing Directory], [Designate Additional Option when Executing], and [Use Output Window].

When [Title] is changed, the set tool is registered as another tool.

4. Click the [Setup] button.

The dialog asking you whether to change the tool opens.

5. Click the [Yes] button.

■ Example of Tool Setup

-When notepad is used

Title	:	note pad
Execute Filename	:	note pad.exe
Option	:	%f
Executing Directory	:	%x
Designate Additional Option when Executing:		A check mark is not set.
Use Output Window	:	A check mark is not set.

CHAPTER 4 MENUS

-When the dir command is registered

Title	:	Dir
Execute Filename	:	command.com
Option	:	/c dir
Executing Directory	:	
Designate Additional Option when Executing:		A check mark is not set.
Use Output Window	:	A check mark is set.

4.7.6 Keyboard

"Keyboard..." enables definition of shortcut keys.

■ Keyboard Setup Procedure

1. Select a type.
Functions are displayed in [Function List].
2. Select the function to be set from [Function List].
The explanation of the selected function is displayed in the explanation field (lower part) of the keyboard setup dialog. When an assigned function is selected, the currently assigned keys are displayed in [Assign key].
3. Set a focus in [New Assign], then specify the key to be assigned to the selected function from the keyboard (press the key).
4. Click the [Set] button.

■ Procedure for Deleting an Assigned Key

1. Select the function corresponding to the key to be deleted (see (1) and (2) in the keyboard setup procedure above).
The currently assigned keys are displayed in [Assign key].
2. Select the key to be deleted from the key list displayed in [Assign key].
3. Click the [Delete] button.
The dialog for checking to be deleted the key opens.
4. Click the [OK] button.

■ Procedure for Changing an Assigned Key

1. Delete an assigned key (see the procedure for deleting an assigned key above).
2. Set a focus in [New Assign], then specify the key to be assigned to the function from the keyboard (press the key).
3. Click the [Set] button.

■ Displaying the Current Setup State List

Click the [Definition List] button to display the key definition list.

■ Restoring all the Set Keys to the Initial State

Click the [Reset] button.

Note:

- Several keys can be assigned to one function. In this case, the assigned keys have the same function.
 - Once the [Set] or [Reset] button is clicked, the set or reset key cannot be canceled. If the [Set] or [Reset] button is clicked by mistake, set the key again.
 - As for the keys that can be set, see Table 4.7-2.
-

Figure 4.7-29 Key Setup Dialog

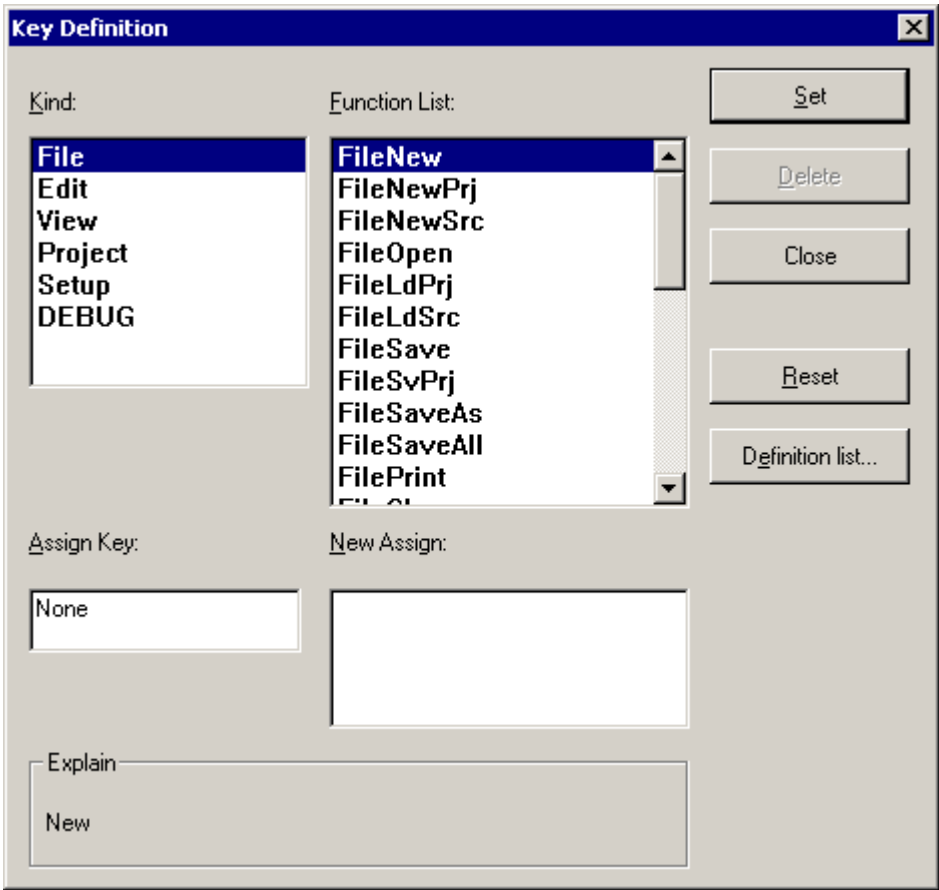


Table 4.7-2 Keys That can be Set

Key	Explanation
CTRL + A to Z	Press any of the A to Z keys while holding down the CTRL key.
SHIFT + CTRL + A	Press any of the A to Z keys while holding down the SIFT and CTRL keys.
SHIFT + F1	Press the F1 key while holding down the SHIFT key.

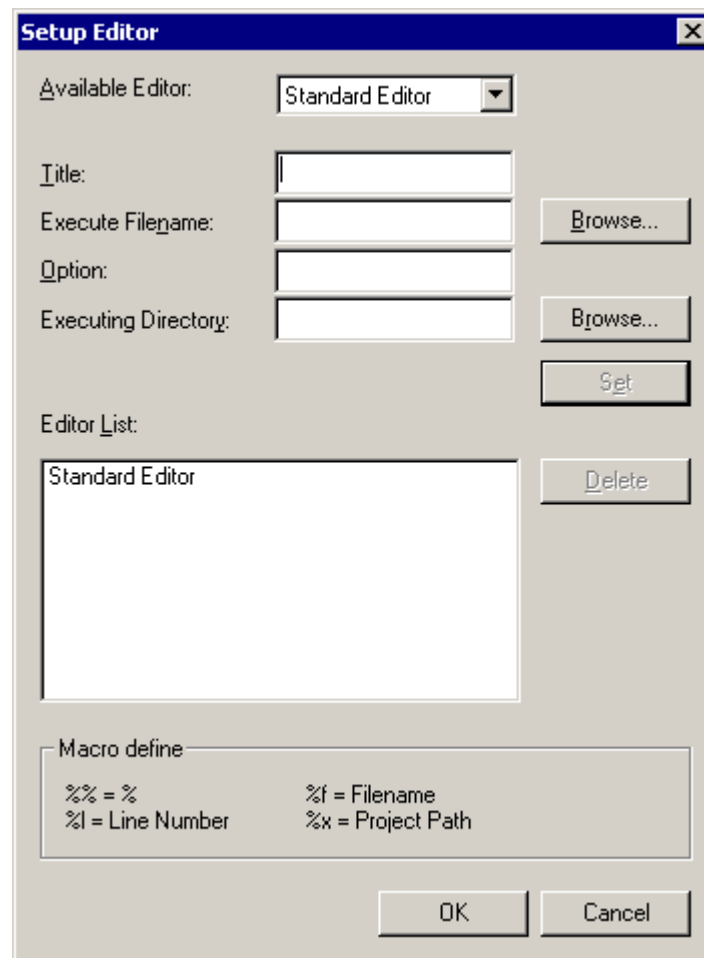
4.7.7 Editor

"Editor..." enables any editor to be registered and used as the standard editor.

■ Registering an Editor

Register the editor to be used instead of the standard editor built into SOFTUNE Workbench in advance. Set the registered editor as the SOFTUNE Workbench editor before editing the file actually. Of the registered editors, the editor set in [Available Editor] is used to edit the file.

Figure 4.7-30 Editor Setup Dialog



■ Editor Registration Procedure

1. Select [Editor...] from the [Setup] menu.
The editor setup dialog shown in Figure 4.7-30 opens.
2. Set a unique title that differs from the registered names.
3. Specify the execution file name of the editor to be registered.
Clicking the [Browse] button to the right of this field opens the file selection dialog. The execution file name of the editor can be selected from this dialog.
4. Set an option.

Macro description can be used in this field. For macro description, refer to Section "1.9 Registering External Editors" in SOFTUNE Workbench User's Manual.

5. Write a run-time directory.

This description may be omitted if control need not to be moved to any specific run-time directory.

Clicking the [Browse] button to the right of this field opens the file selection dialog. A run-time directory can be selected from this dialog.

6. Click the [Set] button.

■ Editor Deletion Procedure

1. Select [Editor...] from the [Setup] menu.
The editor setup dialog shown in Figure 4.7-30 opens.
2. Select the title of the editor to be deleted from the editor list.
3. Click the [Delete] button.

■ Editor Change Procedure

1. Select [Editor...] from the [Setup] menu.
The editor setup dialog shown in Figure 4.7-30 opens.
2. Select the title of the editor to be changed from the editor list.
The values set in [Title], [Execute Filename], [Option], and [Executing Directory] are displayed.
3. Change [Execute Filename], [Option], and [Executing Directory]. When [Title] is changed, the set editor is registered as another editor.
4. Click the [Set] button.
The dialog asking you whether to change the editor opens.
5. Click the [Yes] button.

■ Setting the Editor to be Used

1. Register the external editor to be used according to the editor registration procedure.
2. Click the [▼] button to the right of the [Available Editor] field. The drop-down list showing registered editor titles is displayed.
3. Select the editor title to be used from the drop-down list.

■ Example

Example of Fujitsu Power EDITOR setup

```
Title           : Power EDITOR
Execution Filename : c:\Powered\powered. exe.
Option          : "%f"-g%l
Executing Directory : %x
```

Entering the above and clicking the [Set] button register Fujitsu Power EDITOR in the editor list.

After registering Fujitsu Power EDITOR, select [Power EDITOR] from [Available Editor] and click the [OK] button.

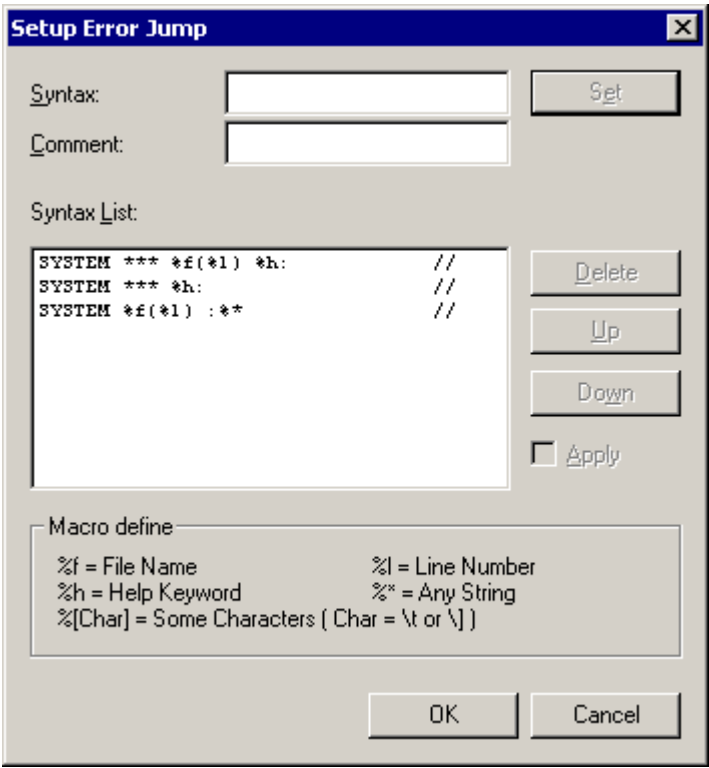
4.7.8 Error

"Error..." registers error message patterns of various tools to enable error jump.

■ Error Jump Setup Procedure

1. Select [Error...] from the [Setup] menu.
The error jump setup dialog shown in Figure 4.7-31 opens.
2. Enter a syntax.
For details on syntaxes, refer to Section "1.7 Error Jump Function" in SOFTUNE Workbench User's Manual.
3. Enter a comment as required.
A comment can be added to each syntax.
4. Click the [Set] button.

Figure 4.7-31 Error Jump Setup Dialog



■ Syntax Deletion Procedure

1. Select [Error...] from the [Set] menu.
The error jump setup dialog shown in Figure 4.7-31 opens.
2. Select the syntax to be deleted from the syntax list.
3. Click the [Delete] button.

■ Syntax Modification Procedure

Modify a set syntax in the following procedure:

1. Select [Error...] from the [Set] menu.
The error jump setup dialog shown in Figure 4.7-31 opens.
2. Select the syntax to modified from the syntax list.
The syntax and comment are displayed in the associated fields.
3. Modify the syntax and comment, then click the [Set] button.
The modified syntax and comment are newly set.
4. Delete an unnecessary syntax (syntax used before modification).

■ Analysis Order Change and Application ON/OFF

-Analysis order change

Error messages are analyzed from the patterns registered in the upper part of the syntax list. To assure correct analysis, the analysis order may have to be changed. The analysis order can be changed in the following procedure:

1. Select [Error...] from the [Set] menu.
The error jump setup dialog shown in Figure 4.7-31 opens.
2. Select the syntax whose order is to be changed from the syntax list.
3. Click the [UP] or [Down] button to move the cursor to the position where error jump is to be set.

-Application ON/OFF

When the check mark to the left of [Apply] is not set, error messages are not analyzed according to the registered syntax.

When a check mark is set to the left of [Apply], ON is displayed in the syntax list. When a check mark is not set, OFF is displayed in the syntax list.

■ Example of Error Jump Setup

When the error format is [error-message: line-number file-name]

Syntax	:	%* : %1 %f
Comment	:	sample

Note:

The syntax for which SYSTEM is displayed in the syntax list cannot be deleted.

4.7.9 Tool Startup

This section explains how to start a registered external tool.

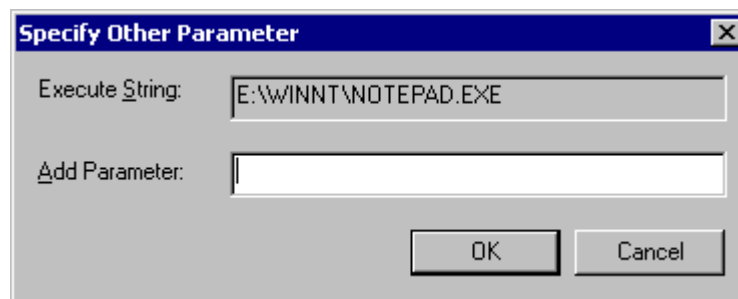
■ Tool Startup

The tools set by Section "4.7.5 Tool" are registered in the submenu. A tool can be started by selecting it from this submenu.

Setting a check mark to the left of [Designate Additional Option when Executing] in tool setup opens the additional option setup dialog shown in Figure 4.7-32 before the tool is started. Set an additional option from this dialog, then click the [OK] button.

The option specified from this dialog is added after the option specified in tool setup and the tool is started.

Figure 4.7-32 Additional Option Setup Dialog



4.8 Window

"Window" controls window display.

■ Control Related to Window Display

- Cascade
- Vertical
- Horizon
- Split
- Arrange Icons
- Refresh
- Refresh All Windows
- Close All Windows

■ Window Name Display

Up to 9 currently open window names are displayed, including icon windows. If ten windows or more are opened, the tenth and subsequent windows are displayed in [Other Windows].

4.8.1 Cascade, Vertical, Horizon

"Cascade", "Vertical", and "Horizon" specify the display formats of subwindows (e.g., Source Window, Register Window, and Assembly Window).

■ Cascade

"Cascade" displays currently open subwindows in cascade.

■ Vertical

"Vertical" arranges currently open subwindows vertically and fully displays them in the main window.

■ Horizon

"Horizon" arranges currently open subwindows horizontally and fully displays them in the main window.

4.8.2 Split

"Split" specifies where a window is vertically split.

■ Split

"Split" specifies where a window is vertically split. The following windows can be vertically split.

- Source Window
- Assembly Window
- Trace Window
- Memory Window
- Coverage Window

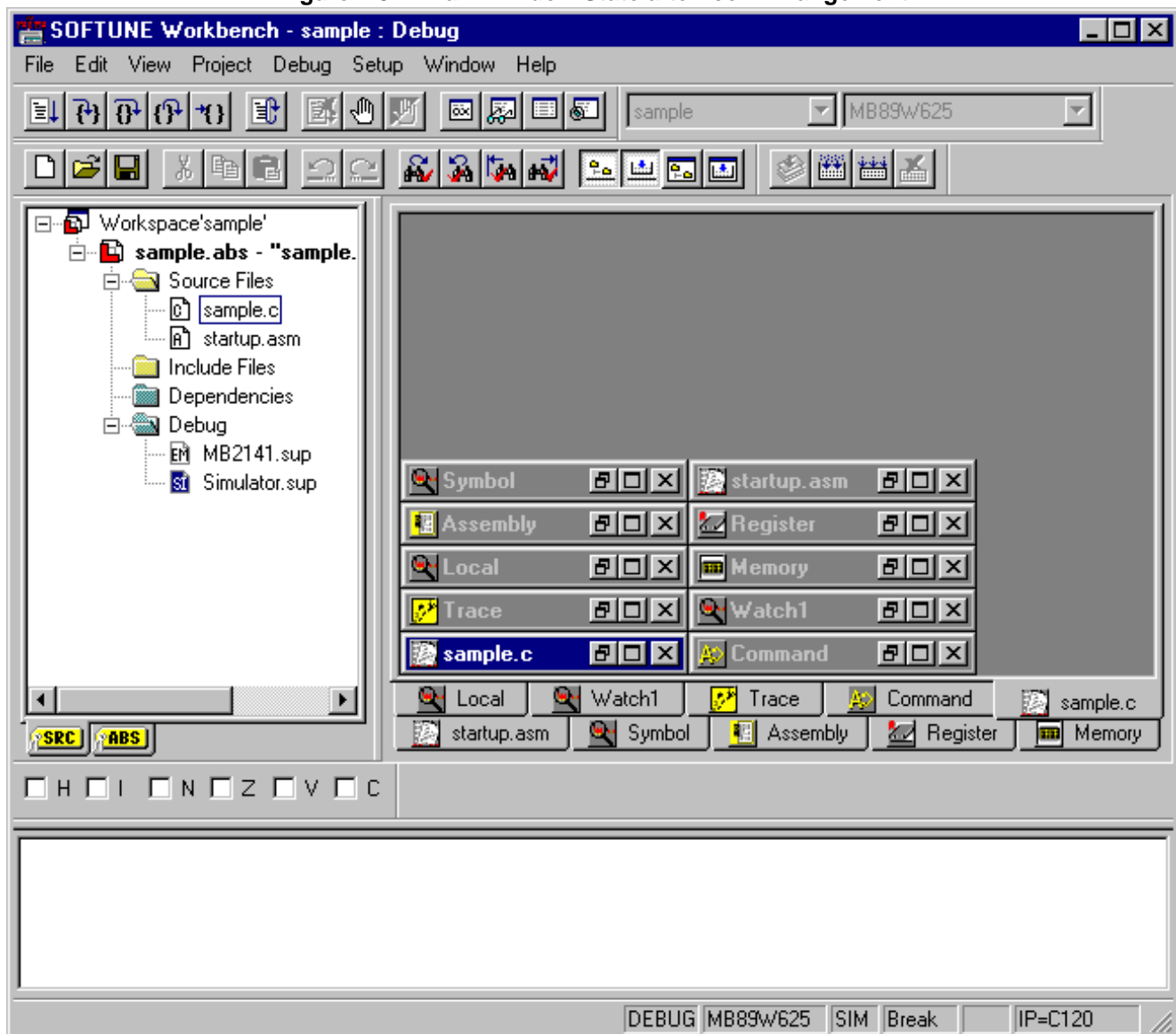
4.8.3 Arrange Icons

"Arrange icons" arranges the locations of the minimized window icons.

■ Arrange Icons

"Arrange icons" arranges all the minimized windows in the SOFTUNE Workbench main window (Figure 4.8-1). However, unminimized windows are not affected.

Figure 4.8-1 Main Window State after Icon Arrangement



4.8.4 Refresh

This command updates information on an active window.

■ Refresh

Information on the current active window is updated.

4.8.5 Refresh All Windows

This command updates information on all the open windows.

■ Refresh all Windows

Information on all the open windows except the SRC tab of project and Output windows is updated.

4.8.6 Close All Windows

"Close all windows" closes all open windows.

■ Close all Windows

"Close all windows" closes all currently open windows other than the Project and Output Windows. If the file edit window being edited has not saved yet, the dialog asking you whether to save the window opens.

The file opened by the external editor cannot be closed by this function.

Even if all windows are closed, the SOFTUNE Workbench state remains unchanged. For this reason, register values, etc., are not affected even during debugging.

4.9 Help

"Help" displays online help.

■ Online Help

- Help Topics
- Support Information

■ Version Information

- Version Information

4.9.1 Help Topics

"Help Topics" retrieves help items according to keywords.

■ Contents

"Contents" hierarchically displays online help contents. It is used to search the contents for the item to be searched.

■ Keyword

SOFTUNE Workbench searches the help file for the directly specified item.

4.9.2 Support Information

"Support Information" opens the attached support information file in the Edit Window.

■ Support Information

Support information provides the information not written in the attached manual. Please read through support information once before using SOFTUNE Workbench.

4.9.3 Version Information

"About Fs896s..." displays SOFTUNE Workbench version information.

■ Version Information Display when SOFTUNE Workbench is not in the Debug Session

"About Fs896s..." opens the version information dialog showing the SOFTUNE Workbench logo mark and version number.

■ Version Information Display when SOFTUNE Workbench is in the Debug Session

"About Fs896s..." opens the version information dialog showing the SOFTUNE Workbench logo mark, version information, type of the currently selected debugger, and type of the target MCU being debugged.

Reference:

Be sure to let us know the displayed version number when asking our company about SOFTUNE Workbench.

CHAPTER 5

Add-in Module

This chapter explains SOFTUNE Workbench Add-in module.

5.1 Customize Bar

5.1 Customize Bar

This chapter describes the Customize Bar of the SOFTUNE Workbench.

5.1.1 What is Customize Bar?

5.1.2 Customize Bar Menu

5.1.3 Registering in the Customize Bar

5.1.4 Warning and Error Messages

5.1.5 Note

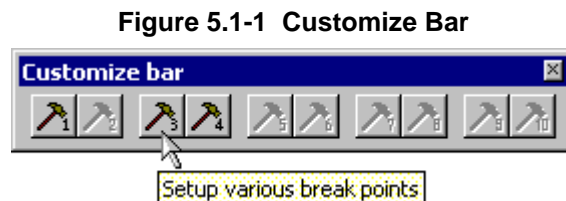
5.1.1 What is Customize Bar?

This section describes the customize bar and how to install it.

■ What is Customize Bar?

The customize bar registers batch file, Workbench menus and external tools used while running the Debugger in a tool bar. This function enables you to call them up easily by clicking one of the buttons.

You can register up to ten objects. Items that have been set once are restored when the Workspace is opened.



By registering batch files, Workbench menus or external tools in the customize bar, the button of the registered number is enabled when the Workspace is opened.

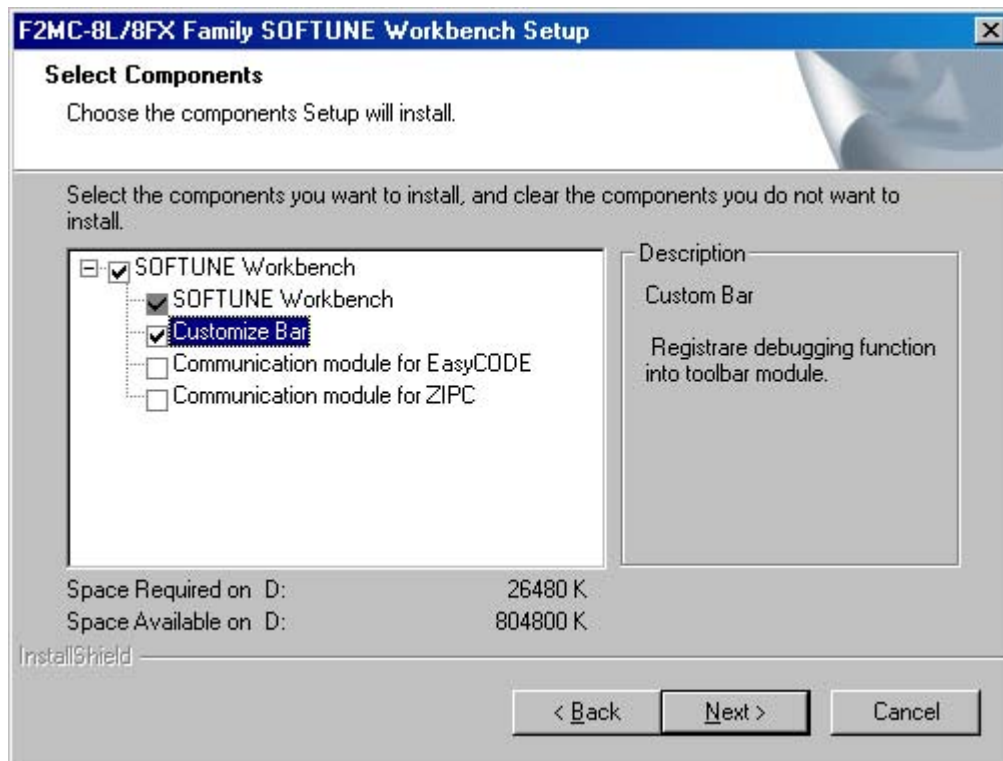
Also, currently registered items are displayed by moving the mouse cursor over any button in the tool bar.

■ Installing Customize Bar

Apply a check mark to the "Customize Bar" in the dialog (Figure 5.1-2) that is displayed when installing SOFTUNE Workbench to install the customize bar.

Note that you can also install only the customize bar if it was not installed when you installed SOFTUNE Workbench.

Figure 5.1-2 Dialog Displayed When Installing



When the customize bar is installed, the "Customize Bar" is added to the SOFTUNE Workbench [View] menu (Figure 5.1-3) and a tool bar (Figure 5.1-1) for the customize bar is displayed.

5.1.2 Customize Bar Menu

This section describes the customize bar menu.

■ Customize Bar Menu

There are two submenus in the [Customize Bar].

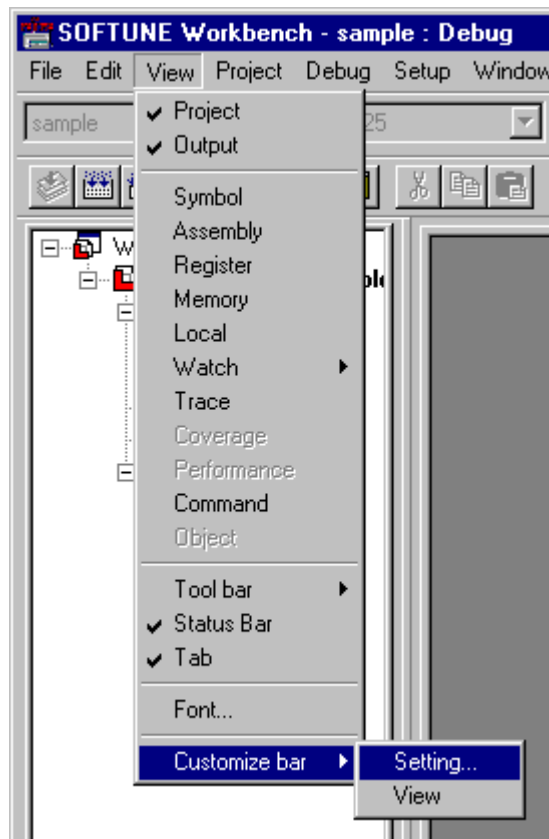
Setting: Registers batch files and Workbench menus in the customize bar.

This menu is enabled when opening a workspace.

View: Switches to view/hide the tool bar for the customize bar.

This menu is always enabled when SOFTUNE Workbench is running.

Figure 5.1-3 Customize Bar Menu



5.1.3 Registering in the Customize Bar

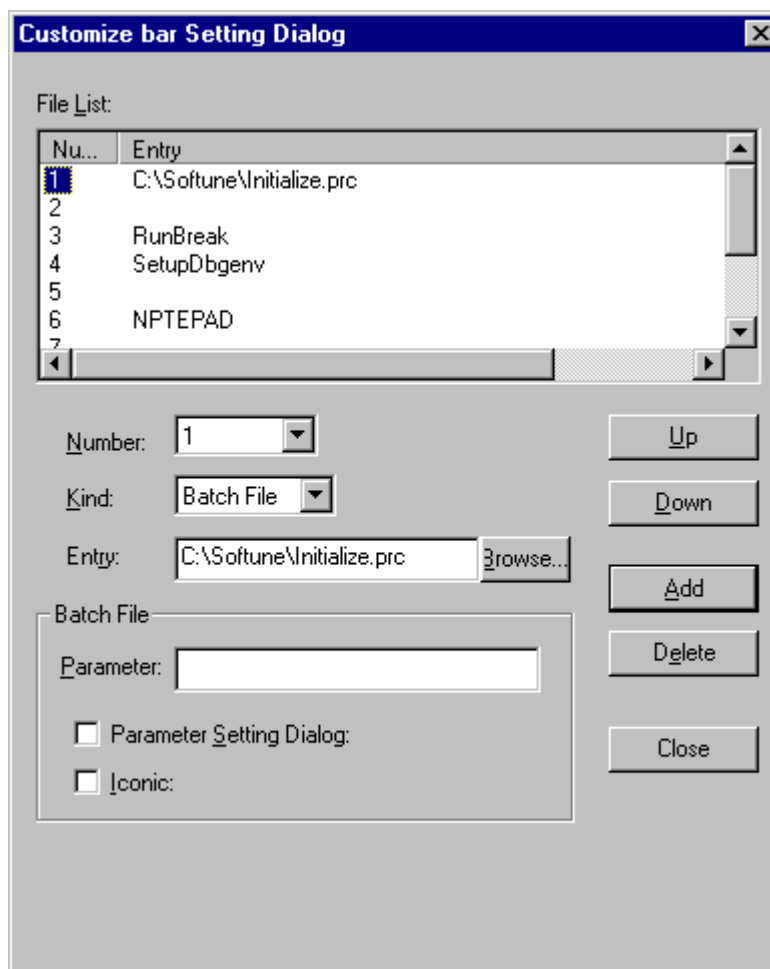
This section describes registering in the customize bar.

■ Registering in Customize Bar

You can registers "batch files", "Workbench menus" and "external tools" in the customize bar. Register using the "Customize Bar Setting Dialog" (Figure 5.1-4) displayed in [View] - [Customize Bar] - [Setting].

For details on how to register, see Sections "5.1.3.1 Registering Batch File", "5.1.3.2 Registering Workbench Menu" and "5.1.3.3 Registering External Tool".

Figure 5.1-4 Customize Bar Setting Dialog



■ Items in Dialog

- File List

The settings registered in the customize bar are displayed.
- Number

Specify the number to register in the customize bar. Numbers from 1 to 10 can be specified.
- Kind

Select either "Batch File" or "Menu" to register in the customize bar.
- Entry
 1. When "Batch File" is selected in "Kind"

Enter the batch file name to be registered in the customize bar. To select from a list, specify with the "Open File" Dialog (Figure 5.1-5) displayed when you click the reference button on the right.
 2. When "Menu" is selected in "Kind"

Enter the Workbench menu to be registered in the customize bar. Specify with the "Menu List" Dialog (Figure 5.1-7) displayed when you click the reference button on the right.
 3. When "Tool" is selected in "Kind"

Enter the external tool to be registered in the customize bar. Specify with the "Select Tool" Dialog (Figure 5.1-8) displayed when you click the reference button on the right.
- Parameter (Enabled only when "Batch File" selected in "Kind")

Input the parameter for executing a batch file.
- Parameter Setting Dialog (Enabled only when "Batch File" selected in "Kind")

Displays "Parameter Setting Dialog" (Figure 5.1-6) that can set/change parameters with the customize bar when executing a batch file.
- Iconic (Enabled only when "Batch File" selected in "Kind")

This makes Workbench an icon when executing a batch file with the customize bar.
- Function Explain (Enabled only when "Menu" selected in "Kind")

Displays a description of the Workbench menu to be registered in the customize bar.
- Up

Changes the order of the registered contents displayed in the "File List" to one above. Switches that order when there is already one registered to a number one above.
- Down

Changes the order of the registered contents displayed in the "File List" to one below. Switches that order when there is already one registered to a number one below.
- Add

Adds batch files, Workbench menu or External tool to the customize bar. If an item has already been registered in the specified number, that number will be rewritten and registered.
- Delete

Deletes the contents registered in the customize bar. Specify the number to delete in the "File List".

Figure 5.1-5 Dialog for Open File

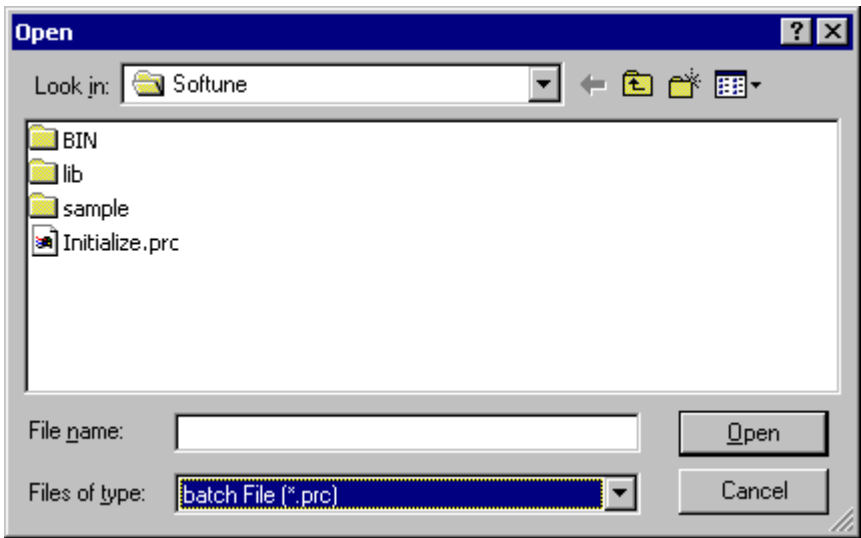


Figure 5.1-6 Parameter Setting Dialog

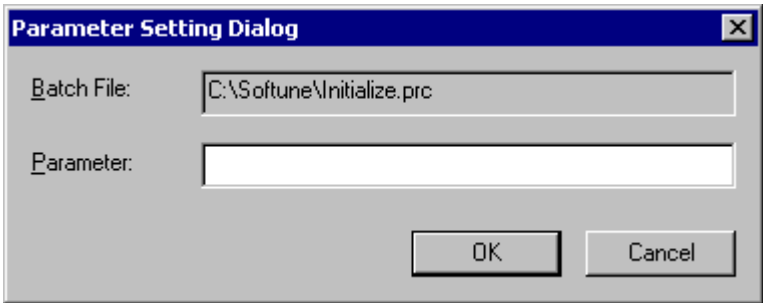


Figure 5.1-7 Menu List Dialog

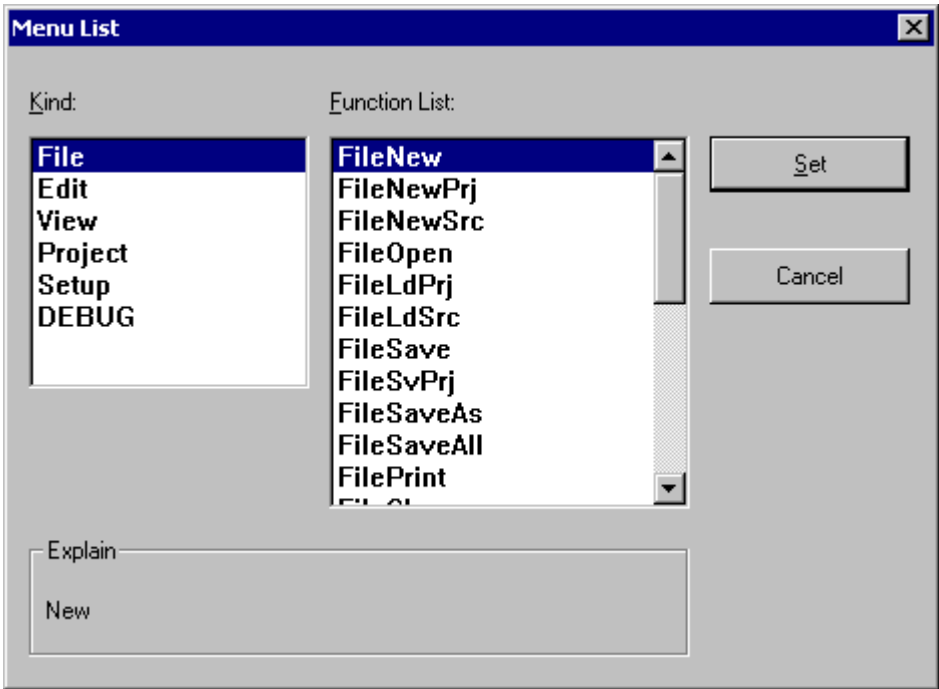


Figure 5.1-8 Select Tool Dialog

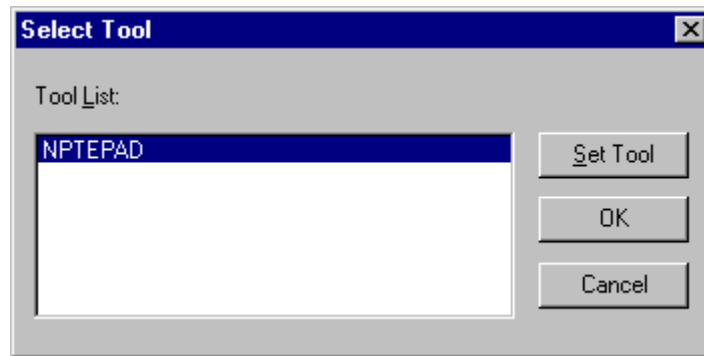
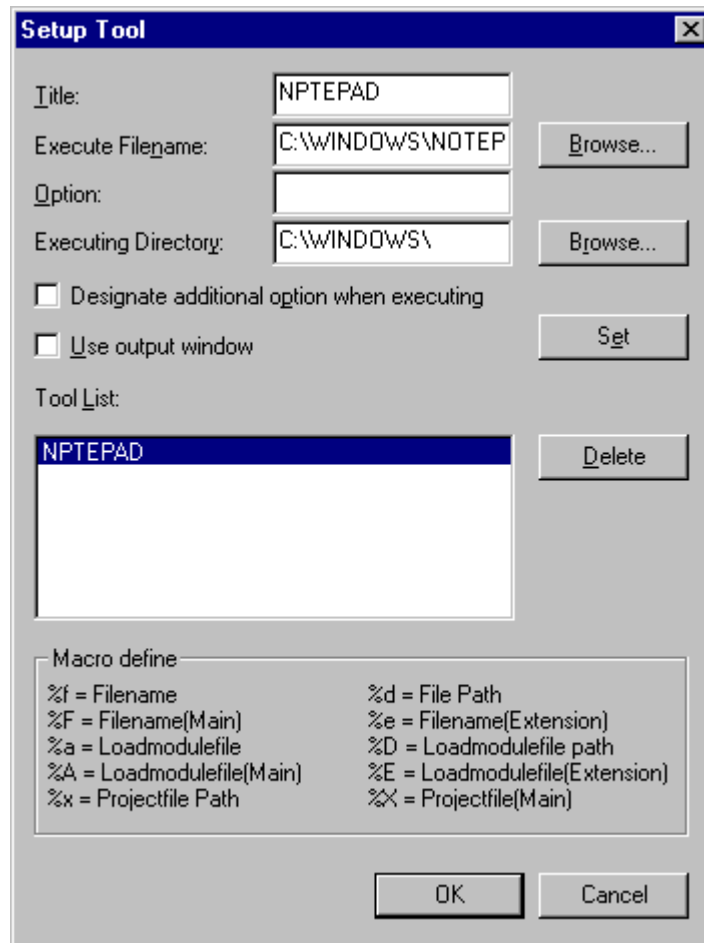


Figure 5.1-9 Setup Tool Dialog



5.1.3.1 Registering Batch File

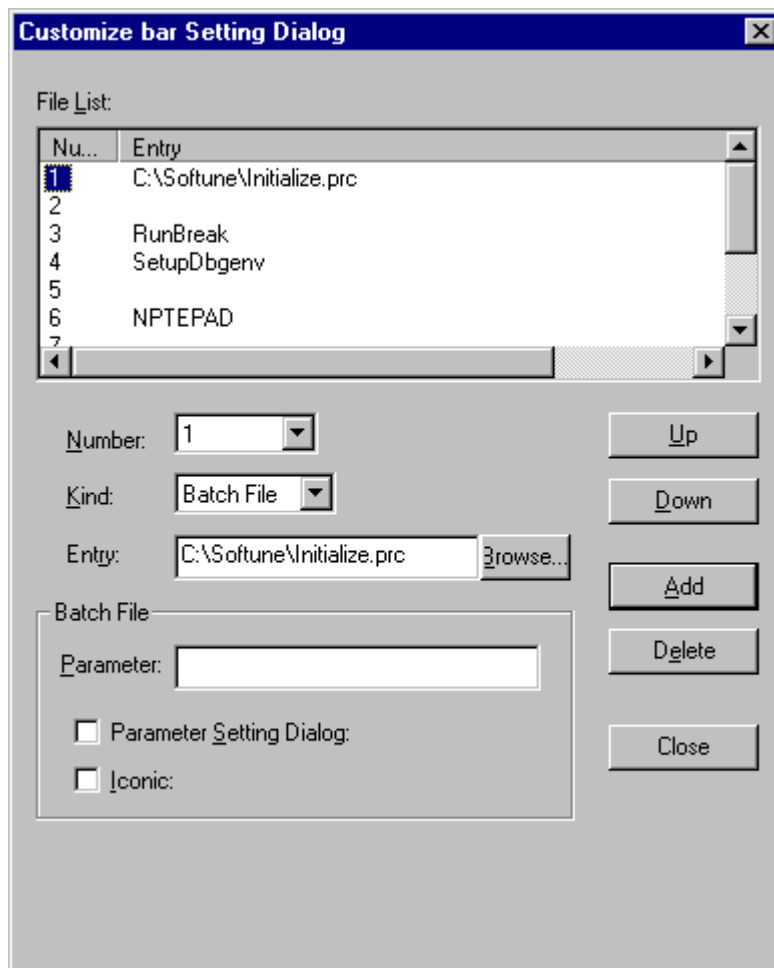
This section describes registering a batch file in the customize bar.

■ How to Register Batch File

1. Display dialog
Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog (Figure 5.1-10).
2. Number
Select the number to be registered in the customize bar.
3. Kind
Select "Batch File".
4. Entry
Enter the batch file name to register in the customize bar. You can specify using the "Open File" dialog (Figure 5.1-5) displayed when you click the reference button on the right.
5. Parameter
Input here when specifying a parameter for a specified batch file. Use a comma to separate multiple parameters.
6. Display the "Parameter Setting" dialog (Figure 5.1-6).
To set a parameter when executing a batch file with the customize bar, apply a check mark to this. This is convenient when you want to specify/change parameters each time you execute a batch file.
7. Iconic
To make Workbench an icon when executing a batch file with the customize bar, apply a check mark to this.
8. Add
Check the input contents of 2 to 7. If they are correct, click "Add".

This completes the registration of a batch file. The contents of the registration are displayed in "File list" and are restored when the Workbench is opened.

Figure 5.1-10 Setting Dialog - Batch File



5.1.3.2 Registering Workbench Menu

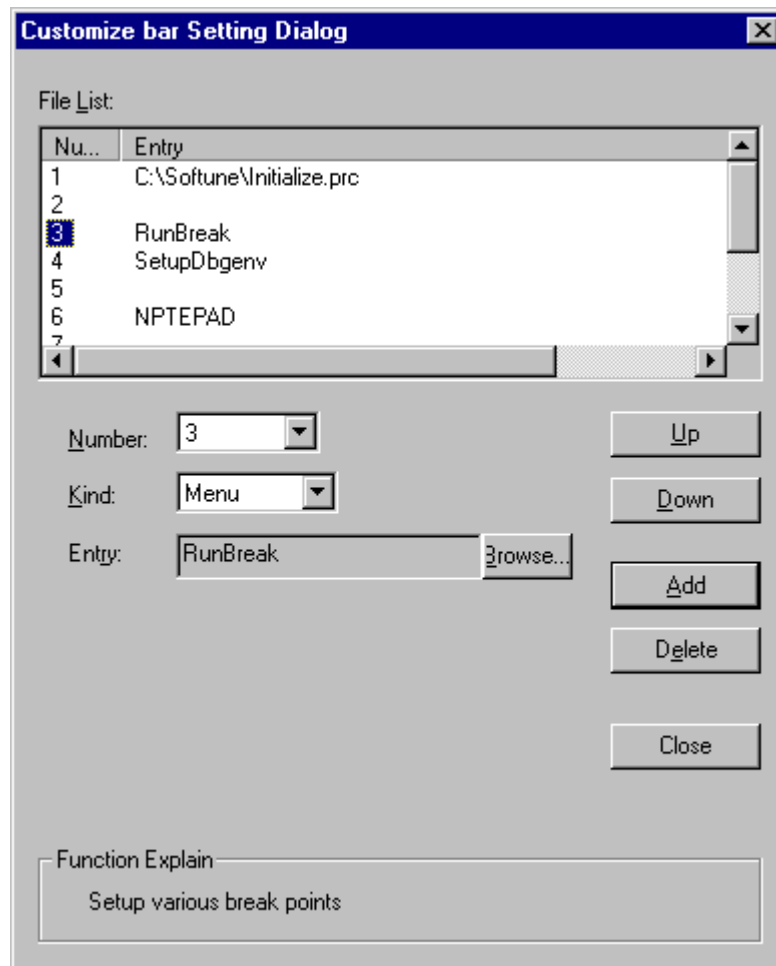
This section describes how to register the Workbench menu in the customize bar.

■ How to Register Workbench Menu

1. Display the dialog
Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog (Figure 5.1-11).
2. Number
Select the number to be registered in the customize bar.
3. Kind
Select "Menu".
4. Entry
Enter the Workbench menu to register in the customize bar. Specify with the "Menu List" dialog (Figure 5.1-7) displayed when you click the reference button on the right.
5. Add
Check the input contents of 2 to 4. If they are correct, click "Add".

This completes the registration of Workbench menus. The registered contents are displayed in "File List" and are restored when the Workspace is opened.

Figure 5.1-11 Setting Dialog - Menu



5.1.3.3 Registering External Tool

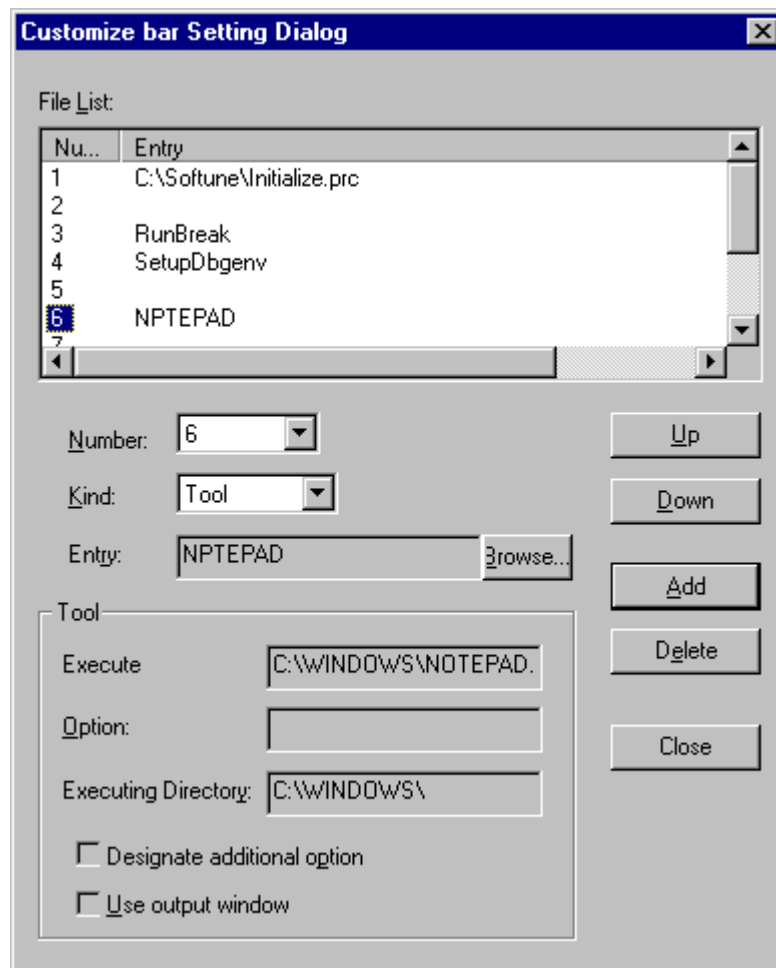
This section describes how to register the external tool in the customize bar.

■ How to Register External Tool

1. Display the dialog
Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog (Figure 5.1-12).
2. Number
Select the number to be registered in the customize bar.
3. Kind
Select "Tool".
4. Entry
Enter the external tool to be registered in the customize bar. Specify with the "Select Tool" Dialog (Figure 5.1-8) displayed when you click the reference button on the right.
Select tool dialog shows the contents tools that are set with the "Set Tool" Dialog (Figure 5.1-9) displayed when you select [Environment] - [Setup tool] menu. Alternatively, you can display "Set Tool" Dialog by the "Set tools" button in the "Select Tool" Dialog.
5. Add
Check the input contents of 2 to 4. If they are correct, click "Add".

This completes the registration of the external tools. The registered contents are displayed in "File list" and are restored when the Workspace is opened.

Figure 5.1-12 Setting Dialog - External Tool



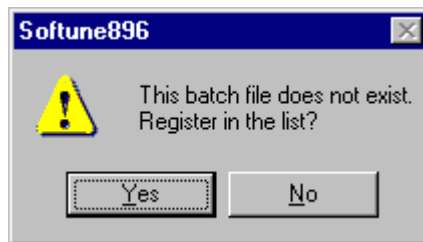
5.1.4 Warning and Error Messages

This section describes the warning and error messages displayed when using the customize bar.

■ Warning Message

1. The following warning message (Figure 5.1-13) is displayed when you click "Add" regardless of whether an input batch file does not exist in the "Entry", when registering a batch file in the customize bar. (See Section "5.1.3.1 Registering Batch File".)

Figure 5.1-13 Warning Message



■ Error Messages

1. The following error message (Figure 5.1-14) is displayed when you click "Add" without entering a batch file in the "Entry", when registering a batch file in the customize bar. (See Section "5.1.3.1 Registering Batch File".)

Figure 5.1-14 Error Message 1



2. The following error message (Figure 5.1-15) is displayed when you click "Add" without entering a machine name in the "Entry", when registering the Workbench menus in the customize bar. (See Section "5.1.3.2 Registering Workbench Menu".)

Figure 5.1-15 Error Message 2

3. The following error message (Figure 5.1-16) is displayed when you click "Add" without entering a tool name in the "Entry", when registering the External tools in the customize bar (See Section "5.1.3.3 Registering External Tool".)

Figure 5.1-16 Error Message 3

5.1.5 Note

This section describes the precautions for using the customize bar.

■ Note

1. The customize bar cannot be used when the Workspace is not opening. When opening the Workspace, the previous settings are restored and the customize bar buttons are enabled.
2. Customized bars for registration are stored for each tool. This allows a common set of restored data to be provided even if projects, workspaces or setup files are switched.
3. When quitting the Debugger, and the setup file was not saved, the registered contents of the customize bar will not be saved. For that reason, when restarting the Debugger, the registered contents of the customize bar will not be restored.
4. When registering batch files in the customize bar, input the relative path or the absolute path from the current directory (= project directory) for the batch files.
5. When registering batch files in the customize bar, always use a comma to separate parameters when specifying multiple parameters. This is the same for the "Parameter Setting" dialog (Figure 5.1-6).
5. If you have registered either a batch file or a Workbench menu to be enabled with the Debugger started in the customize bar, pressing the corresponding button does not execute the registered item unless the Debugger has been started.

APPENDIX

The Appendixes describes the register name, downloading monitor program, setting LAN interface, setting USB interface, creating ROM on monitor debugger target, external I/F for simulator.

APPENDIX A List of Register Names

APPENDIX B Downloading Monitor Program

APPENDIX C Setting LAN Interface

APPENDIX D Setting USB Interface

APPENDIX E Installing Monitor Debugger

APPENDIX F External I/F for Simulator

APPENDIX A List of Register Names

Register names are displayed.

■ Registers

The registers that can be operated by SOFTUNE Workbench differ for each MCU type as follows:

Accumulator	: A
Temporary accumulator	: T
Stack pointers	: SP
Program counter	: PC
General-purpose registers	: R0 to R7
Interrupt level	: IL0, IL7
Others	: RP, DP, CCR, PS, IX, EP
Flags	: H, I, N, Z, V, C

APPENDIX B Downloading Monitor Program

To use the emulator debugger, the monitor program corresponding to the chip to be used must be written to the emulator.

This processing is called "monitor program download".

■ When the emulator is the MB2141 series

The Downloading Monitor Program procedure is described below:

1. Connect the emulator to a personal computer (PC) with an RS-232C or LAN interface.
(When connecting the emulator to the LAN, see APPENDIX C Setting LAN Interface.)
2. Press the reset switch, then turn on the emulator.
Check that the READY LED of the emulator body turn on.
3. Execute the [Monitor Loader] menu from [SOFTUNE V3] of [F²MC-8L Family SOFTUNE Workbench Tools] of the Windows Start Menu.
The monitor loader program is started.
4. Select the monitor program to be loaded.
Select the monitor program corresponding to the chip to be used.
5. Specify a communication type.
To use the RS-232C interface, specify a communication port and a baud rate.
To use the LAN interface, specify the host name of the emulator.
6. Click [Start Load].
The selected monitor program is downloaded to the emulator.
7. Select [Exit] from the [File] menu to exit the monitor program.

Table B-1 Monitor Program

Chip type	Corresponding chip	Monitor program
F ² MC-8L	MB896XX	EML96A.HEX(*1) EML96N.HEX(*2) EML96NW.HEX(*3)

*1: For MB2144-505 emulation pod

*2: For MB2144-508 emulation pod

*3: For MB2144-508 emulation pod (new emulator interface)

APPENDIX C Setting LAN Interface

To enable LAN communication, the LAN interface must be set at the PC and emulator sides. Consult the LAN administrator when setting the IP address and a port address, etc.

■ Setting LAN Interface at PC Side

1. Install the TCP/IP protocol in network setting dialog on WindowsXP, Windows2000.
Install the TCP/IP protocol.
Click [Control Panel]-[Network and Dial-up Connections]-[Local Area Connection]-[Property] to set in WindowsXP and Windows2000.
2. Add the IP address, assigned to the emulator, to the HOSTS file.
Add the following items:
IP address Host name
For WindowsXP, Windows2000, the IP address and host name are in SYSTEM32 \DRIVERS\ ETC.
Users with administration authority must set the address and name.
3. Register the emulator port address and service name in the SERVICES file. At the default, 5001 is the support address, and fjicesv is the service name. Register the following items:
fjicesv 5001/tcp
For WindowsXP, Windows2000, the port address and service name exist in SYSTEM32 \DRIVERS\ ETC. Users, who possess an administrator authority, must set these address and name.

■ Setting LAN Interface at Emulator Side

In case of MB2141 emulator, this procedure is following:

1. Connect the emulator to the PC with the RS-232C interface.
2. Turn on the emulator.
3. Execute the [LAN Address] menu from [SOFTUNE V3]- [F²MC-8L Family SOFTUNE Workbench Tool] of the Start Menu.
The LAN address setup program is started.
4. Select emulator name (MB2141).
5. Click [Set Communication] to set the RS-232C interface.
6. Click [Read] to read the current emulator setting status.
7. Set IP Address and Port Address. The IP address and port address set values at the PC side are displayed.
8. Usually, [Universal] is used as MAC Address. However, when using [Local] as MAC Address, consult the LAN address administrator.
9. Select [Exit] from the [Setup] menu to exit the LAN address setup program.
10. Press the emulator reset button (button on rear) to reset the emulator.

APPENDIX D Setting USB Interface

Communication via USB requires installation of the USB driver in the personal computer.

■ Installation of USB Driver

WindowsXP and Windows2000 connect the emulator to a personal computer via USB.

To install the USB driver, proceed as follows:

1. Connect the emulator to a personal computer with the USB cable.
2. When the power supply of the emulator is turned on, the OS requests installation of the USB driver. Specify the directory (Drivers) on this product CD-ROM.

Note:

The following emulator have USB interface.
MB2146-09

APPENDIX E Installing Monitor Debugger

This appendix describes the requirements for using Monitor Debugger.

- Conditions required for the target system
- Including and setting up the monitor program
- Sample programs
- Prohibitions and restrictions

■ Introduction

Monitor Debugger allows you to debug your application program on an intended mass-produced microcontroller with built-in flash memory using a monitor program in combination with the application program. To use Monitor Debugger, add the monitor program main routine (FGM.rel) to the application program to create your target system.

■ System Configuration for Using Monitor Debugger

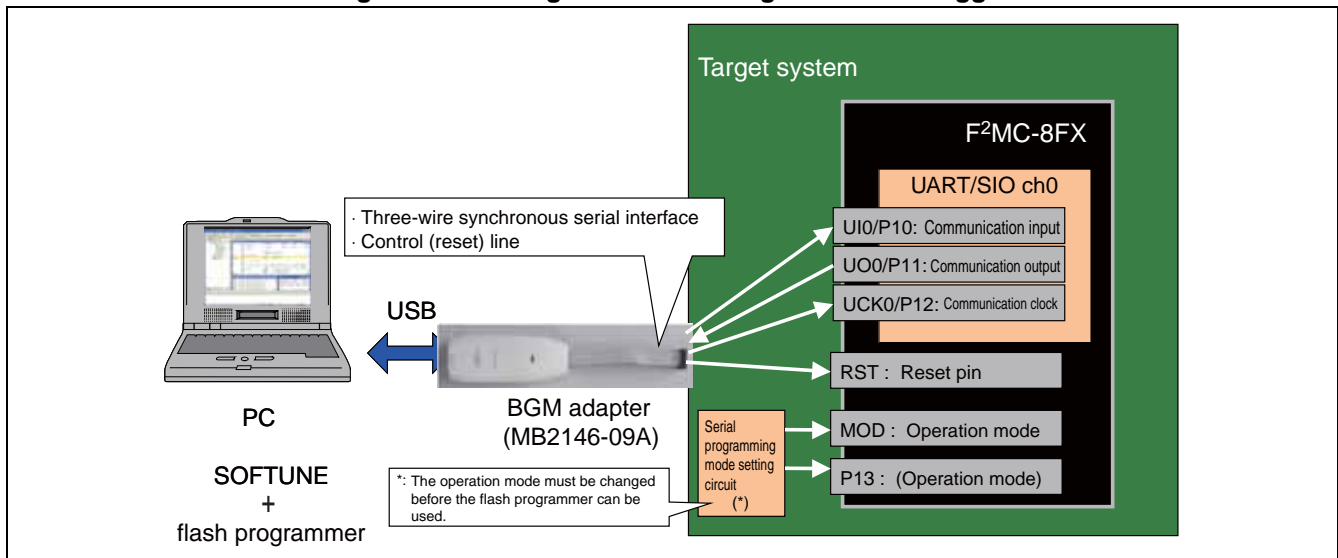
The following configuration is required to use Monitor Debugger:

- SOFTUNE Workbench (V30L30 or later)
- BGM adapter (MB2146-09A)
- Target board: Carrying an 8FX series of microcontroller with built-in flash memory in the F²MC-8FX family

Note also that the following device is useful for programming into flash memory with the microcontroller left on the board.

● Flash programmer

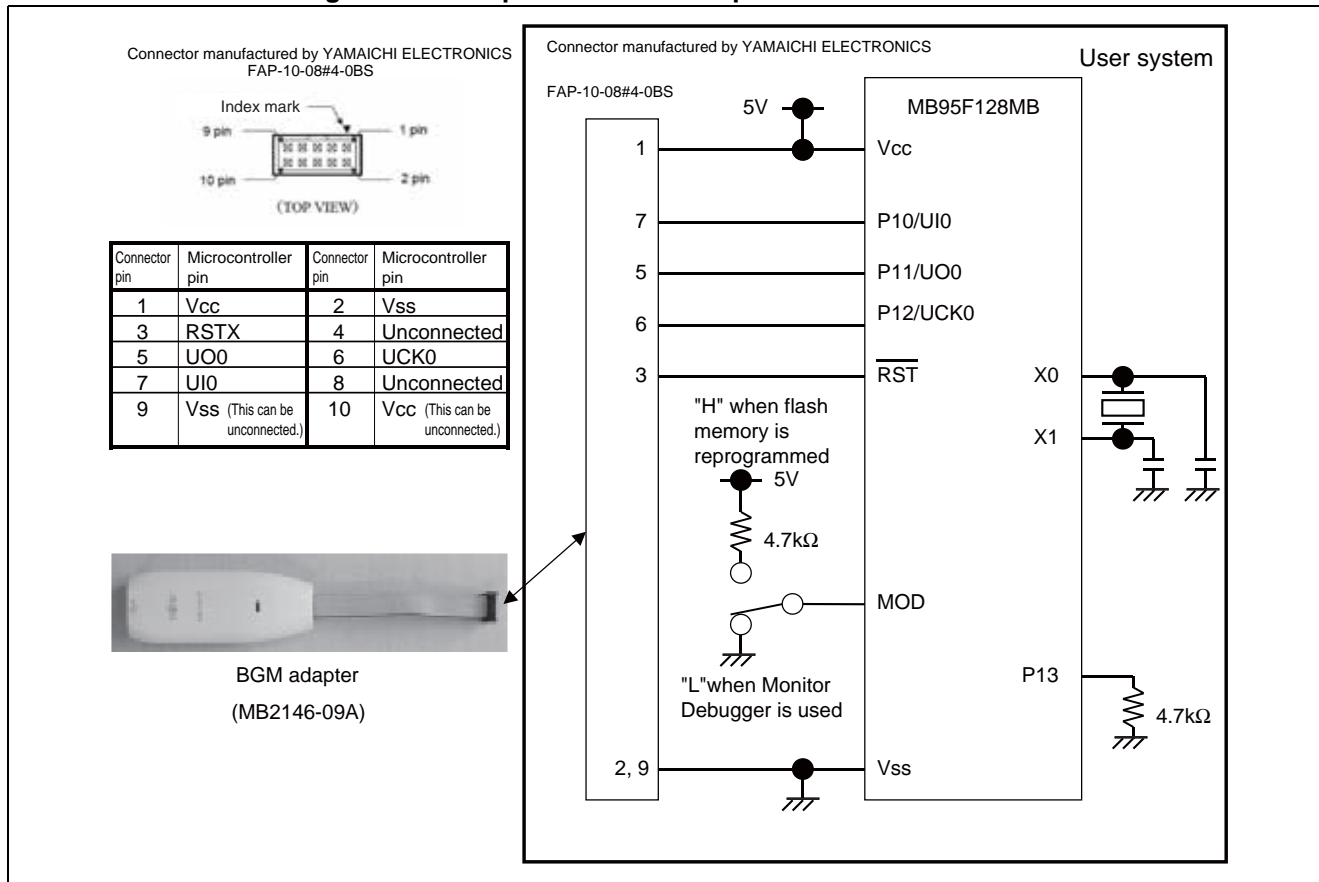
Figure E-1 Configuration for Using Monitor Debugger



To configure the circuit for connecting the BGM adapter to the target system, refer to "sample connections

for serial programming" in the hardware manual for each microcontroller. Monitor Debugger is designed to work with the same connection as that for on-board flash programming, except for the operation mode setting pin.

Figure E-2 Sample Circuit for Adapter Connection Circuit



■ Conditions Required for the Target System

To use Monitor Debugger, the target system must satisfy the following conditions:

- Input clock frequencies

Table E-1 Input Clock Frequencies

Environment	Condition	Remarks
Main oscillator clock frequency	1 MHz or more	
Sub-oscillator clock frequency	32.768 kHz	(Only for dual-clock products)

- The following resources must be able to be allocated for the monitor program:

Table E-2 Resources for Monitor Program

Resource name	Condition	Remarks
CPU core	F ² MC-8FX	
ROM (flash memory)	About 1.5 Kbytes · 0xFA00 - 0xFFBF (*1) · 0xFFC0 - 0xFFC1 (*2) · 0xFFFF2 - 0xFFFF3 (*3) · 0xFFFFC - 0xFFFFF (*4)	Location addresses fixed (Note) *1: Monitor code, etc. *2: CALLV #0 vector *3: UART/SIO interrupt vector *4: Reset vector
RAM	17 byte · 0x0473 - 0x047F · Stack area (4 bytes) (*5)	*5: 4 bytes added to the stack area for application program
UART/SIO	1ch	
RESET pin	Operable via the BGM adapter	
I/O ports	The pins are shared with the UART/SIO. 3 I/O ports (*6)	*6: P10, P11, P12
Wild register	3 points	

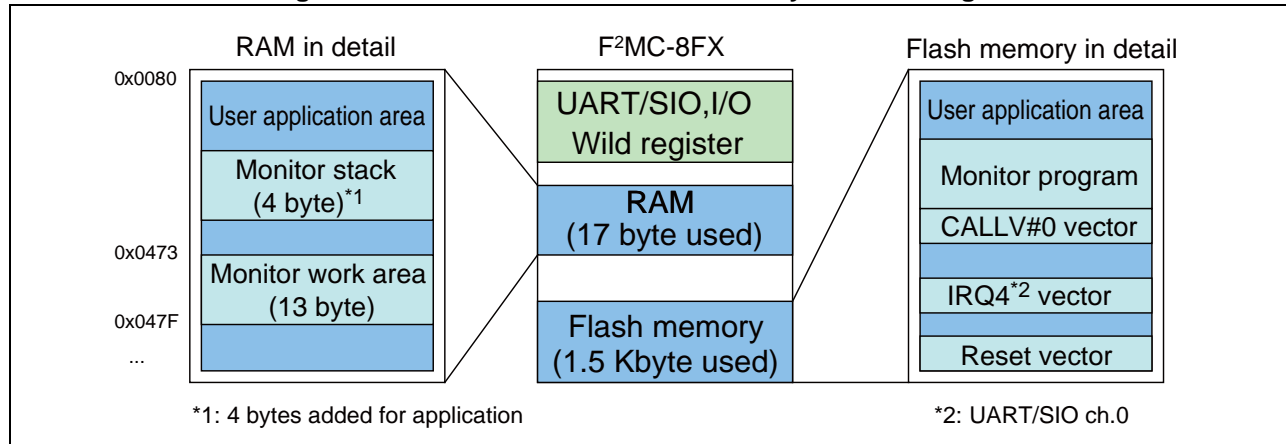
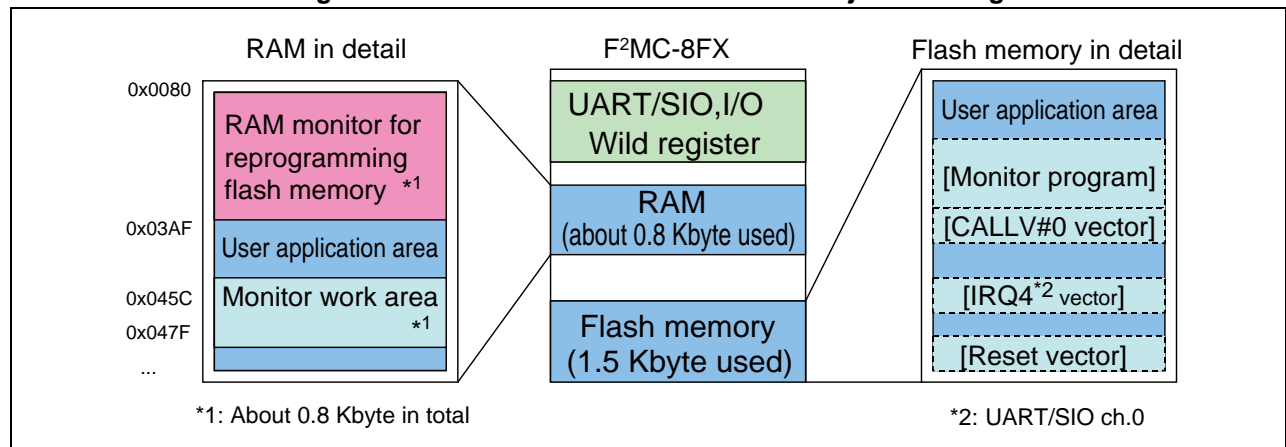
(Note: The start address of monitor code may vary depending on the version of the monitor program.)

- In addition to the above resources, the following resource must be available temporarily during object loading. (Note)

Table E-3 Additional Resource for Object Loading

Resource name	Condition	Remarks
RAM	About 0.8 KByte · 0x0080 - 0x03AF · 0x045C - 0x0472	Location addresses fixed

(Note: The resources used during object loading are requirements for Monitor Debugger. For the requirements for using the flash programmer, refer to the relevant manuals.)

Figure E-3 Outlined Resources for Use by Monitor Program**Figure E-4 Outlined Resources Used for Object Loading**

■ Including and Setting Up the Monitor Program

To use Monitor Debugger, add and modify the monitor program files listed below. In addition, part of the application program must also be modified. See the following sections for details.

■ Changing Monitor Operation Settings (fgm_cfg.h)"

■ Modifying the Application Program"

■ Creating an Auto-boot Checker"

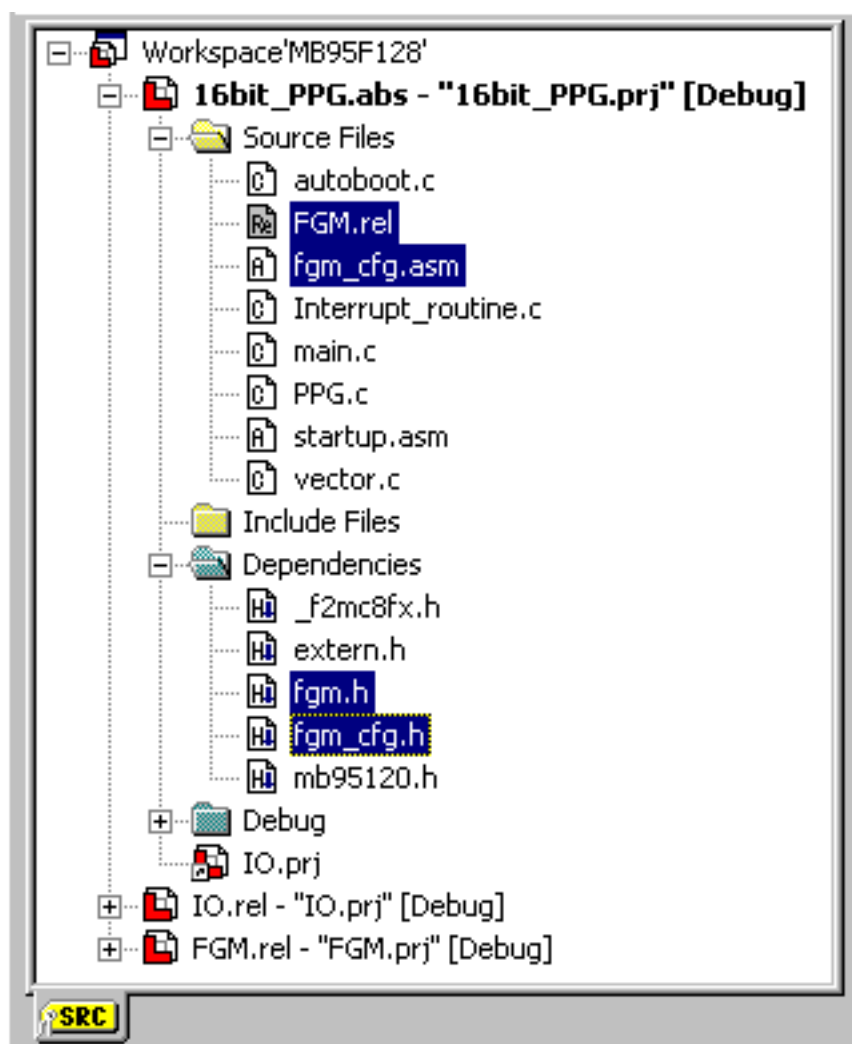
● Additional files

Table E-4 Monitor Program Files to be Added

File name	Description	Remarks
FGM.rel	Monitor program main routine	Source Files
fgm_cfg.asm	Monitor operation definitions	Source Files
fgm.h	Header file for including the monitor program	Dependencies
fgm_cfg.h	Monitor operation setting header	Dependencies must be modified. *1

(*1 : See "■ Changing Monitor Operation Settings (fgm_cfg.h)".)

Figure E-5 Example of Sample Program File Configuration



● Changes to the application program

Table E-5 Changes to the Application Program

Change	Description	Remarks (Function name)
Reset vector	Change this to the monitor program start address.	FGM_INIT()
Stack area	Add 4 bytes to the monitor program area.	
Watchdog timer start routine	Replace this with the monitor program API.	FGM_WDTON()
Interrupt level setting register	Allow UART/SIO interrupt requests to be accepted.	Bits 1 and 0 in the ILR1 register
UART/SIO interrupt vector definition	Register the UART/SIO interrupt service to the vector.	FGM_INT()
Addition of auto-boot checker	Add the process of checking for auto-booting.	*2

(*2: See "■ Creating an Auto-boot Checker".)

■ **Changing Monitor Operation Settings (fgm_cfg.h)**

To use Monitor Debugger, you have to correct the settings of the following items to suit the application program. The set values will be macro-defined (#define) to be included as the operation definition table via the monitor operation definition file (fgm_cfg.asm).

If you use symbols in place of numeric values, they must be able to be interpreted by the assembler. (For a C language function of "AutoBootCheck()", for example, use "_AutoBootCheck" instead.)

Figure E-6 Coding Example in fgm_cfg.h

```

=====
; F2MC-8FX Family Monitor Debugger configuration file,
; ALL RIGHTS RESERVED, COPYRIGHT (C) FUJITSU LIMITED 2007
; LICENSED MATERIAL - PROGRAM PROPERTY OF FUJITSU LIMITED
=====
/* Start address */
    .IMPORT __start
#define USER_STARTADR __start .....(1)

/* Stack top */
    .IMPORT STACK_TOP
#define USER_STACKTOP STACK_TOP .....(2)

/* Auto-boot address */
    .IMPORT _AutoBootCheck
#define USER_ABOOTADR _AutoBootCheck .....(3)

/* password (4byte) */
#define USER_PASSWORD 0x12345678 .....(4)

```

Table E-6 Monitor Operation Setting Items

Item	Description	Remarks
USER_STARTADR	Application program start address	(1)
USER_STACKTOP	Stack pointer initial value for the application program	(2)
USER_ABOOTADR	Auto-boot checker start address	(3)
USER_PASSWORD	Password for connecting the debugger	(4)

● USER_STARTADR

Specify the start address of the application program. This address is set as the reset vector when the monitor program is not included. If you specify a symbol, correct the argument of the IMPORT instruction at the same time.

Figure E-6 (1) is an example in which the start address is "__start".

● USER_STACKTOP

Specify the address of the stack area to be used for the application program.

For the stack area, allocate the size of space for use by the application program and additional 4 bytes for use by the monitor program.

If you specify a symbol, correct the argument of the IMPORT instruction at the same time.

Figure E-6 (2) is an example in which the stack area address is "STACK_TOP".

● USER_ABOOTADR

Specify the address of the auto-boot checker. The auto-boot checker is described in "■ Creating an Auto-boot Checker". If you specify a symbol, correct the argument of the IMPORT instruction at the same time.

Note that setting this address to 0x0000 or 0xFFFF makes the auto-boot checker invalid.

Figure E-6 (3) is an example in which the auto-boot checker is a C language function of "AutoBootCheck()".

● USER_PASSWORD

Specify the password for connecting the debugger. Set a 4-byte value.

Figure E-6 (4) is an example in which the password is 0x12345678.

Note:

To use the password effectively, use the "flash security feature" of flash memory as well. Enabling the flash security feature protects flash memory from being read from or written to through external pins while allowing debugging with Monitor Debugger. Use meticulous care in password management.

■ Modifying the Application Program

Modify the application program to include the monitor program.

The table below lists the items to be modified.

Table E-7 Items to Be Changed in the Application Program

Item	Description	Remarks
Change to reset vector	Change this to the monitor program start address.	FGM_INIT()
Stack area	Add a 4-byte area for the monitor program.	
Watchdog timer start routine	Replace this with the monitor program API.	FGM_WDTON()
Interrupt level setting	Allow UART/SIO interrupt requests to be accepted.	Bit1 and bit0 in the ILR1 register
UART/SIO interrupt vector definition	Register the UART/SIO interrupt service to the vector.	FGM_INT()
Addition of auto-boot checker	Add the process of checking for auto-booting.	

● Change to reset vector

Set the reset vector to the monitor program start address "FGM_INIT()".

"FGM_INIT()" is defined in "fgm.h". Specify "FGM_INIT" in C or "_FGM_INIT" in the assembler.

Figure E-7 Coding Example in C

```
#include "fgm.h"
(omission)
#pragma section CONST=RESETVECT, locate=0xfffc
void (* const reset_vector[2])()={
    (void (*)())(0x00), /* 0xfffc,d = 0x0000    */
    FGM_INIT          /* 0xfffe,f = "FGM_INIT" */
};
```

Figure E-8 Coding Example in the Assembler

```
#include "fgm.h"
(omission)
.section RESETVECT, CONST, LOCATE=0xFFFC
.data.h 0x0000 ; 0xfffc,d
.data.h _FGM_INIT ; 0xfffe,f
```

● Stack area

Add 4 bytes as the area for the monitor program.

In the stack pointer register, set the bottom address of the stack area obtained with 4 bytes added. Use the same value to define "USER_STACKTOP" in "fgm_cfg.h".

Figure E-9 Coding Example in the Assembler

```

;-----
; definition to stack area
;-----
        .SECTION  STACK,  STACK,  ALIGN=1
        .RES.B   30      ; Application Program
        .RES.B   4       ; Monitor Debugger
STACK_TOP:
(omission)
        .SECTIONCODE,  CODE,  ALIGN=1
__start:
;-----
; set stack pointer
;-----
        MOVW    A, #STACK_TOP
        MOVW    SP, A

```

● Watchdog timer start routine

If the watchdog timer is used, replace the routine for starting it first after releasing the microcontroller from a reset with a monitor program API of "FGM_WDTON()". Using this API can prevent the watchdog timer from generating a reset even when control is passed to the monitor program. Note that the argument of the API is set in the WDTC register.

This API can be called not only from a source written in C but also from a source in the assembler by stacking the arguments in the stack.

Table E-8 API FGM_WDTON()

Monitor program API	FGM_WDTON
Processing	Starts the watchdog timer
Argument	unsigned char SETVAL Value set in the WDTC register
Return value	void: None
Sample use	
C language	#define SETVAL 0x45 FGM_WDTON(SETVAL);
Assembler	#define SETVAL 0x45 mov a,#SETVAL pushw a call _FGM_WDTON popw a

Figure E-10 Coding Example in C

```

#include "fgm.h"
(omission)
/* IO_WDTC.byte = 0x45; */
FGM_WDTON(0x45); /* Start Watchdog Timer. */

```


Figure E-11 Coding Example in the Assembler

```
#include Åfgm.hÅh
(omission)
/* mov _IO_WDTON,#0x45 */
mov  a,#0x45
pushw a
call _FGM_WDTON
```

Note:

Do not single-step through the API "FGM_WDTON". Otherwise, the watchdog timer is started before the monitor program is set up, involving the risk of generating a reset by the watchdog timer.

- **Interrupt level setting**

Allow UART/SIO interrupt requests to be accepted.

If the ILR1 register is manipulated within the application program, therefore, modify the program such that bit1 and bit0 are set to B'00 (= request interrupt level 0).

If the ILR1 register is not manipulated by the application program, the monitor program sets ILR1 to 0xFC during its internal initialization, eliminating the need for additional setting.

Figure E-12 Coding Example in C

```
/* L07 = 0, L06 = 1, L05 = 2,*/
/* IO_ILR1.byte = 0x1B; */ /* L04 = 3 */
IO_ILR1.byte = 0x18; /* L04 = 0 */
```

Figure E-13 Coding Example in the Assembler

```
/* L07 = 0, L06 = 1, L05 = 2,*/
/* mov _IO_ILR1,#0x1B ; L04 = 3 */
mov _IO_ILR1,#0x18 ; L04 = 0
```

- **UART/SIO interrupt vector definition**

The forced break of Monitor Debugger is implemented by the UART/SIO interrupt routine. The UART/SIO interrupt routine "FGM_INT()" must therefore be registered to the vector.

Figure E-14 Coding Example in C (Excerpt from Sample Program "vector.c")

```
#include "fgm.h"
(omission)

#pragma intvect FGM_INT 4/* UART/SIO ch0 */
```

Figure E-15 Coding Example in the Assembler

```
#include "fgm.h"
(omission)

.section  INTVECT, CONST, LOCATE=0xFFFF2
.data    _FGM_INT    /* IRQ04: UART/SIO ch0 */
```

● Addition of auto-boot checker

After a reset, the monitor program waits for connection from the debugger without starting the application program. (This allows debugging from the beginning of the application program.)

When debugging is completed, however, the application system cannot run independently with the debugger disconnected. To solve this problem, the auto-boot checker is executed immediately after a reset to select whether the application program is started automatically.

Note, however, that the auto-boot checker is not required when the application system does not have to run independently.

The auto-book checking method is not specified precisely. Since it is executed immediately after a reset, pay attention to that resources have not be initialized and that the execution, if it takes time, can delay the start of the application. For details, see "■ Creating an Auto-boot Checker".

■ Creating an Auto-boot Checker

The auto-boot checker is required for the application system to work with the monitor program included after debugging.

Table E-9 Auto-boot Checker

auto-boot checker	(USER_ABOOTADR)
Processing	Checks whether the application can get started automatically.
Argument	void : None
Return value (EP register value)	int : Enable/disable automatic start. 0x0000 = Do not start the application automatically. Else = Start the application automatically.
Notes	SP register: This routine is called with this register containing the value defined in (USER_STACKTOP) in fgm_cfg.h.
	PS (program status) register: Set such that the direct bank pointer specifies 0x400 to 0x47F. This does not affect resource register access to 0x0000 to 0x007F. If access to RAM from 0x0080 to 0x00FF is required, be careful to specify an extended address.

The auto-boot checker is executed immediately after the microcontroller is released from a reset. It must therefore be coded to be executable with resources not initialized or with C-defined initial-valued variables not initialized. Also consider that some system registers are set to fixed values as mentioned above.

Note also that the check process taking longer time delays the application to be started.

● Coding in the C language

Return int as a return value. Return 0x0000 if the auto-boot conditions have not been satisfied. When they have been satisfied, return a value other than 0x0000, such as 0x0001.

Shown below is an example from a sample program.

Figure E-16 Coding Example in C (Excerpt from Sample Program "autoboot.c")

```
int AutoBootCheck(void)
{
    IO_DDR1.bit.P13 = 0; /* P13=Input */

    if (IO_PDR1.bit.P13 == 1){
        return 0x0001; /* AutoBoot */
    } else {
        return 0x0000; /* DEBUG */
    }
}
```

Figure E-17 Setting Example in fgm_cfg.h

```
/* Auto-boot address */
IMPORT _AutoBootCheck
#define USER_ABOOTADR _AutoBootCheck
```

APPENDIX

The above example is coded to check the status of I/O port P13 (*1) and causes an auto-boot when the port is High or waits for connection from the debugger if not.

In "fgm_cfg.h", a symbol following "_" is used as the one that can be referenced by the assembler.

*1: On the MB95F128MB, I/O port P13 must be set along with the MOD pin to program flash memory on the board. (It does not have to be set for object loading from the debugger.)

Note:

Immediately after the power is turned on, the I/O ports may be unstable in status. For referencing an I/O port by the auto-boot checker, design it in due consideration of its electrical characteristics.

● Coding in the assembler

Return the return value to the EP register. Set the EP register to 0x0000 if the auto-boot conditions have not been satisfied. When they have been satisfied, set the EP register to a value other than 0x0000, such as 0x0001.

Shown below is an example in an assembler source.

**Figure E-18 Coding Example in the Assembler
(Assembler Code Having the Same Effect as Sample Program "autoboot.c")**

```
_AsmABootCheck:
    clrb _IO_DDR1:3 /* P13=Input */

    bbc _IO_PDR1:3, debug
    movw EP, #0x0001 /* AutoBoot */
    jmp exit
debug:
    movw EP, #0x0000 /* DEBUG */
exit:
    ret
```

Figure E-19 Setting Example in fgm_cfg.h

```
/* Auto-boot address */
.IMPORT _AsmABootCheck
#define USER_ABOOTADR _AsmABootCheck
```

Like the sample program in the C language, this example is coded to check the status of I/O port P13 (*1) and causes an auto-boot when the port is High or waits for connection from the debugger if not.

In "fgm_cfg.h", a symbol is used as it is as the one that can be referenced by the assembler.

■ Programming into the Target

Monitor Debugger cannot be connected to the microcontroller containing no monitor program. Before attempting to connect the debugger, therefore, program some application program including a monitor program.

● For on-board programming

Refer to the hardware manual for the target microcontroller to prepare a flash programmer.

For on-board programming using the flash programmer, the pins listed below are used on the MB95F128MB as the target of the sample program. For connection of any other microcontroller, refer to the manuals for the flash programmer and the microcontroller.

(*: The pins to be used may vary depending on the flash programmer used.)

Table E-10 Pin Functions for On-board Programming

Pin	Function	Setting example (*) with the flash programmer in used
MOD,P13	Mode pin	MOD = "H", P13 = "L"
X0, X1	Oscillation pins	Input a frequency of at least 1 MHz
RST	Reset pin	(Connected to the flash programmer. The pin must be able to turn on/off the reset.)
P10 / UI0	Serial data input pins	(Connected to the flash programmer)
P11 / UO0	Serial data output pins	(Connected to the flash programmer)
P12 / UCK0	Serial clock input pins	(Connected to the flash programmer)
Vcc	Supply voltage feeder pin	(Connected to the flash programmer. The pin must be supplied with power from the system at a voltage that enables flash memory programming.)
Vss	GND pin	(Connected to the flash programmer)

(*: The settings may vary depending on the flash programmer used.)

● Using a microcontroller with a monitor program already programmed

Additional programming is not required (*1) unless the monitor program is erased.

Note that, except the MOD and P13 pins, the Monitor Debugger connection pin and flash programmer connection pin are common. You should therefore configure your system such that the MOD and P13 pins are reconfigurable.

*1: When the application program is modified, the debugger performs loading.

■ Provided Sample Programs

The following files are provided as the monitor program and sample application program.

Table E-11 List of Provided Sample Programs

Monitor program			
File name		Description	Folder name
m1	fgm_cfg.asm	Monitor operation definitions	16bit_PPG\
m2	FGM.rel	Monitor program main routine	16bit_PPG\
m3	fgm.h	Header file for including the monitor program	16bit_PPG\
m4	fgm_cfg.h	Monitor operation setting header	16bit_PPG\
Sample application programs			
File name		Description	Folder name
a1	startup.asm	Startup routine	16bit_PPG\
a2	main.c	Main routine	16bit_PPG\
a3	PPG.c	16bit PPG resource initialization	16bit_PPG\
a4	Interrupt_routine.c	16bit PPG interrupt routine	16bit_PPG\
a5	vector.c	Interrupt/reset vector definitions	16bit_PPG\
a6	extern.h	Application program header	16bit_PPG\
a7	autoboot.c	Auto-boot checker	16bit_PPG\
Sample I/O register file			
File name		Description	Folder name
r1	_f2mc8fx.c	Sample I/O register file main	IO\

The following source is also attached as a reference material.

Table E-12 Provided Source Code (Reference Material)

Monitor program source (reference)			
File name		Description	Folder name
s1	fgm_main.asm	Monitor program	FGM\

■ Prohibitions and Restrictions

This section summarizes the items to follow and note on debugging with the monitor program included.

● Prohibitions

[Prohibited] Do not manipulate any resource being used by Monitor Debugger.	
	Monitor Debugger may behave unpredictably if you manipulate RAM or a UART/SIO register being used by Monitor Debugger. When using the I/O ports such as P13 to P15, in particular, be careful not to change the settings of P10 to P12.

[Prohibited] Do not manipulate the PLLC or SYCC register via the debugger.	
	Monitor Debugger adjusts the communication speed based on the values of the PLLC and SYCC registers. If you manipulate the PLLC or SYCC register directly from the memory window or watch window, the monitor program and Monitor Debugger cannot detect the change and thus cannot readjust the communication speed. This may disable further communication and operation from the debugger. (There is no problem with changing the settings of the PLLC and SYCC registers within the application program.)

[Prohibited] Do not set a break point within the monitor program.	
	Even if a break point is set, the monitor program does not break normally due to its principles of operation. In addition, it cannot guarantee its subsequent behavior.

[Prohibited] Do not single-step through API FGM_WDTON.	
	Doing so causes the watchdog timer to get started before the monitor program is set up, involving the risk of generating a reset by the watchdog timer.

● Restrictions:

[Restricted] The initial value of the SP register is changed.	
	<p>Originally, the SP register is initialized to 0x0000 immediately after a reset. Since the monitor program uses the stack area, however, the application program is started with the register containing a value other than 0x0000.</p> <p>Be sure to re-set the value within the application program irrespective of the set value.</p>

[Restricted] The time of starting after a reset is cleared is changed.	
	<p>Even when auto-booting has been selected, the application program is started after the monitor program initialization routine and auto-boot checker are executed. Pay attention to the timing after a reset is cleared, which is important to the application program.</p>

[Restricted] A forced break is not available while UART/SIO interrupts are disabled.	
	<p>A forced break during the execution of the application is implemented by the UART/SIO interrupt routine. A forced break cannot be used to halt the application when the interrupt enable flag (I) is 0 or while an interrupt whose priority is higher than UART/SIO interrupts is being serviced. (Be careful not to mistake Monitor Debugger operation errors for such breaks.)</p>

[Restricted] Dual-clock products (with a sub clock input) require longer response time.	
	<p>The speed of communication with a microcontroller which can operate with the sub clock may decrease to around 128 bps. Immediately after a code break, forced break, or single-stepping, in particular, the settings of the PLLC and SYCC registers may have been updated and thus the communication starts at around 128 bps.</p> <p>If the microcontroller is operating actually with the sub clock, the communication remains at a speed from 128 bps to 4000 bps. The monitor program slows down its operation and response time as well.</p> <p>(Be careful not to mistake Monitor Debugger operation errors for such delays.)</p>

[Restricted] Code breaks are invalid during step-in execution.

When Monitor Debugger steps in for each machine instruction, it uses a wild register temporarily. In that period, no code break can be set, leaving code break points invalid. Be careful in particular if a break point has been set within an interrupt routine with the "interrupt mask" disabled during single-stepping. As the break point within the interrupt routine is invalid, no break takes place even when the break point service is executed.

[Restricted] Add 4 bytes as the stack area for the monitor program.

When the application program passes control to the monitor program, 4 bytes in the stack area are used to save the application program break address and PS register value. If the stack area is insufficient, the break address and PS register value cannot be obtained correctly, preventing subsequent debugging from being performed normally.

[Restricted] Use the "flash security feature" as well.

To use the password effectively, use the "flash security feature" of flash memory as well. Enabling the flash security feature protects flash memory from being read from or written to through external pins while allowing debugging with Monitor Debugger. (Erasing is possible.) Use meticulous care in password management. If you have forgotten the password, use the flash programmer for erasing and then program a password-known monitor program into flash memory.

[Restricted] To change the password, use the flash programmer.

If the debugger loads an application whose password has been changed, the debugger terminates with an error due to password discrepancy. If you have changed the password, use the flash programmer to ensure the programming of the application.

[Restricted] A reset occurs after object loading.

For object loading, the RAM monitor is used to reprogram flash memory. After reprogramming, Monitor Debugger issues a reset for transition from the RAM monitor to the regular monitor. Note also that RAM values recorded before object loading are destroyed.

[Restricted] Consider the electrical characteristics observed immediately after the power is turned on.

Immediately after the power is turned on, the I/O ports may be unstable in status. For referencing an I/O port by the auto-boot checker, design it in due consideration of its electrical characteristics.
--

■ If Monitor Debugger Loses Control

If the application program runs out of control, for example, Monitor Debugger may lose control (communication is disabled between the host system and the target).

In that case, follow the procedure below to restart the debugger:

1. Select "Debug (D)" - "Abort (A)".
(Otherwise, press the execution stop button.)
2. Press "Abort (A)" in the abort dialog.
*: The MCU cannot be reset at this time.
Ignore recurrent display.
3. When the "Cannot Abort" warning is issued, press "OK".
4. Terminate the debugger and reset the target system.
5. Restart the debugger.

APPENDIX F External I/F for Simulator

Simulator debugger for F²MC-8L/8FX supports the external I/F to create peripheral simulation modules. This section describes external I/F function.

■ Outline of External I/F DLL

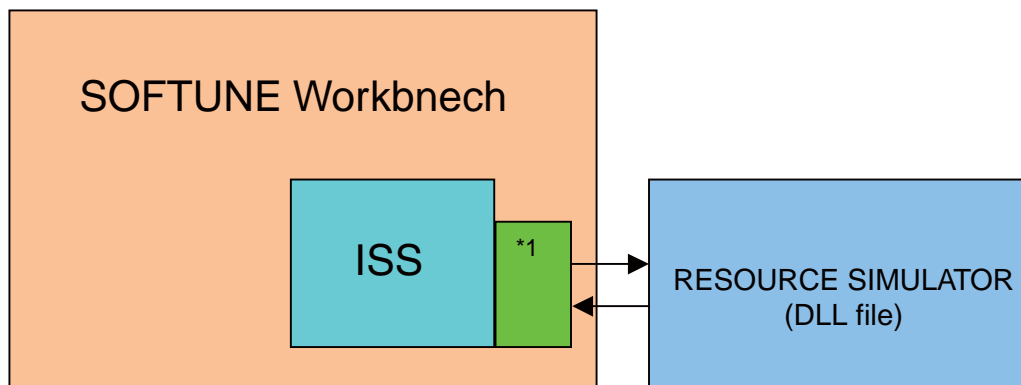
The Simulator Debugger for SOFTUNE Workbench supports the I/O simulation function (ports/interrupts) for aiding in debugging applications using microcontroller resources.

However, this function is simple so that it cannot cover the simulation of complicated microcontroller resources; therefore, providing interface between the instruction set simulator (ISS) and a microcontroller resource simulator that the user describes in the C language.

This microcontroller resource simulator is created by the dynamic link library (DLL) file for Windows.

■ Configuration

Figure F-1 External I/F Module



The microcontroller resource simulator is read only when the Simulator Debugger for SOFTUNE Workbench is started.

The interface between the ISS and the microcontroller resource simulator is called in the following timing:

- When the Debugger is started
- When the Debugger has been initialized.
- When the Debugger is terminated
- When the target is reset
- Immediately before data is read from memory (I/O)
- Immediately after data is written to memory (I/O)
- Immediately before fetching is performed
- When an interrupt occurs
- When a timer event occurs

There are the following functions for operating the ISS from the microcontroller resource simulator:

- Reads/writes data from/to memory

- Reads/writes data from/to registers
- Sets interrupt sources
- Requests abort of instruction execution

■ Simulator External I/F Specification

[Function List]

ISS -> DLL

1. SSDI_Entry (Startup)
2. SSDI_Init (Initialize)
3. SSDI_Ready (Ready)
4. SSDI_End (End)
5. SSDI_Reset_Event (Post Reset)
6. SSDI_Read_Event (Post Read)
7. SSDI_Write_Event (Post Write)
8. SSDI_Execute_Event (Post Execute Instruction)
9. SSDI_Interrupt_Event (Post Interrupt)
10. SSDI_Timer_Event (Post Timer)

DLL -> ISS

11. SSDI_Read_Memory (Read from Memory)
12. SSDI_Write_Memory (Write to Memory)
13. SSDI_Read_Register (Read from Register)
14. SSDI_Write_Register (Write to Register)
15. SSDI_Set_Interrupt (Set Interrupt Source)
16. SSDI_Set_Timer (Set Timer)
17. SSDI_Request_Abort (Request Abort)
18. SSDI_Set_Area (Set Area)
19. SSDI_Clear_Cycle (Clear cycle count)
20. SSDI_Execute_MCU (Execute target program)
21. SSDI_GetCurrentExecuteAddr (Get the Address in Execution)

F.1 SSDI_Entry (Start)

[Format]

```
int SSDI_Entry (int id, HINSTANCE hInstance)
```

[Argument]

int	id;	// DLL ID
HINSTANCE	hInstance	// External I/F module handle

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger.

[Explanation]

A call is made when the debugger is started.

Perform necessary initialization.

[Remarks]

DLL ID (id) always indicates 0.

The external I/F module handle (hInstance) is used to obtain an ISS entry.

At this point, no external I/F module can be called.

(Example)

```
typedef int (WINAPI *LPFNREADMEMORY)();
LPFNREADMEMORY Read_Memory;
extern "C" int WINAPI SSDI_Entry (int id, HINSTANCE if)
{
    int my_dll_id = id;
    FARPROC pF_Read_Memory = ::GetProcAddress(if, "SSDI_Read_Memory");
    Read_Memory = (LPFNREADMEMORY)pF_Read_Memory;
}

int func()
{
    if (ReadMemory(addr, size, len, data) != len)
        error();
}
```

F.2 SSDI_Init (Initialize)

[Format]

int SSDI_Init (void)

[Argument]

None

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger.

[Explanation]

A call is made when the debugger is started.

Perform necessary initialization.

[Remarks]

Now an external I/F module can be called.

F.3 SSDI_Ready (Ready)

[Format]

int SSDI_Ready (void)

[Argument]

None

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger.

[Explanation]

A call is made when the debugger is started.

[Remarks]

It is called after having done batch file execution and an automatic load in debugger start.

F.4 SSDI_End (End)

[Format]

void SSDI_End (void)

[Argument]

None

[Return value]

None

[Explanation]

A call is made when the debugger is ended.

Perform necessary end processing.

[Remarks]

F.5 SSDI_Reset_Event (Post Reset)

[Format]

void SSDI_Reset_Event (void)

[Argument]

None

[Return value]

None

[Explanation]

Posts a reset of the debugger by a command or the issue of a reset.

Initializes resources.

[Remarks]

Sets all the interrupt states OFF.

F.6 SSDI_Read_Event (Post Read)

[Format]

```
int SSDI_Read_Event (READ_EVENT *info)
```

[Argument]

```
typedef struct {
    unsigned long   addr;           // Access address
    int             size;           // Access size (1:Byte / 2:Word / 4:LONG)
    unsigned long   total_cycle;    // Total cycle count
    unsigned long   inst_cycle;     // Count of cycles from beginning of instruction
    unsigned long   *data;          // Read data
    unsigned long   *cycle;         // Count of cycles taken for access
    ORDER_TYPE      type;           // Instruction type notification
} READ_EVENT;

typedef enum {
    ORDER_NORMAL,    // Other than read-modify instructions
    ORDER_RMW        // Read-modify instruction
} ORDER_TYPE;
```

[Return value]

```
=0          Read data enabled
!=0         Read data disabled
```

[Explanation]

Posts the occurrence of a read access event by instruction execution.

When read data is enabled, the ISS operates assuming read data (info.data) as read data. When read data reflected in ISS memory, data must be written by using `SSDI_Write_Memory()`.

When read data is disabled, data is read from ISS memory.

Set the count of cycles taken for read access as the count of cycles taken for access (info.cycle).

If this access cause an error, use `SSDI_Request_Abort()` to stop instruction execution.

[Remarks]

The count of cycles from the beginning of the instruction (inst_cycle) is always posted as 0.

F.7 SSDI_Write_Event (Post Write)

[Format]

```
int SSDI_Write_Event (WRITE_EVENT *info)
```

[Argument]

```
typedef struct {
    unsigned long   addr;           // Access address
    int             size;           // Access size (1:Byte / 2:Word / 4:LONG)
    unsigned long   data;           // Write data
    unsigned long   total_cycle;    // Total cycle count
    unsigned long   inst_cycle;     // Count of cycles from beginning of instruction
    unsigned long   *cycle;         // Count of cycles taken for access
    ORDER_TYPE      type;           // Instruction type notification
} WRITE_EVENT;

typedef enum {
    ORDER_NORMAL,           // Other than read-modify instructions
    ORDER_RMW               // Read-modify instruction
} ORDER_TYPE;
```

[Return value]

```
=0           Write data enables
!=0          Write data disabled
```

[Explanation]

Posts the occurrence of a write access event by instruction execution.

When write data is enabled, the ISS does not write data to memory. Therefore, when write data is reflected in ISS memory, data must be written by using `SSDI_Write_Memory()`.

When write data is disabled, data is written to ISS memory.

Set the count of cycles taken for write access as the count of cycles taken for access (`info.cycle`).

If this access cause an error, use `SSDI_Request_Abort()` to stop instruction execution.

[Remarks]

The count of cycles from the beginning of the instruction (`inst_cycle`) is always posted as 0.

F.8 SSDI_Execute_Event (Post Execute Instruction)

[Format]

```
void SSDI_Execute_Event (EXECUTE_EVENT *info)
```

[Argument]

```
typedef struct {  
    unsigned long    addr;           // Access address  
    unsigned long    total_cycle;    // Total cycle count  
} EXECUTE_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of a fetch access event by instruction execution. This event posts only the starting address of the instruction immediately before instruction execution.

[Remarks]

F.9 SSDI_Interrupt_Event (Post Interrupt)

[Format]

```
void SSDI_Interrupt_Event (INTERRUPT_EVENT *info)
```

[Argument]

```
typedef struct {  
    int          int_number;    // Interrupt number  
    unsigned long total_cycle;   // Total cycle count  
    unsigned long int_cycle;     // Cycle count of interrupt processing  
    unsigned long *cycle;       // Count of cycles delayed  
} INTERRUPT_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of an interrupt event by instruction execution.

Set the count of cycles generated during processing as the count of cycles delayed (info.cycle). In the ISS, this value is added to the cycle count.

If this event causes an error, stop instruction execution by using SSDI_Request_Abort().

[Remarks]

The cycle count of interrupt processing (int_cycle) is always posted as 0.

F.10 SSDI_Timer_Event (Post Timer)

[Format]

```
void SSDI_Timer_Event (TIMER_EVENT *info)
```

[Argument]

```
typedef struct {
    unsigned long    total_cycle;    // Total cycle count
    unsigned long    inst_cycle;     // Differential cycle count from previous event
    int              time_id;        // Set timer ID
    unsigned long    *cycle;         // Count of cycles delayed
} TIMER_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of a timer event set in the ISS.

Set the count of cycles generated during processing as the count of cycles delayed (info.cycle). In the ISS, this value is added to the cycle count.

If this event causes an error, stop instruction execution by using SSDI_Request_Abort().

[Remarks]

Set the timer event conditions in SSDI_Set_Timer().

F.11 SSDI_Read_Memory (Read from Memory)

[Format]

int SSDI_Read_Memory (unsigned long addr, int size, int length, void *data)

[Argument]

unsigned long	addr;	// Access address
int	size;	// Access size (1:Byte / 2:Word / 4:LONG)
int	length;	// Read data count
void	*data;	// Data storage area

[Return value]

Count of data completely read.

If the return value is different from the read data count (length), an error occurs.

[Explanation]

Reads data from memory.

The type of the data storage area (data) differs depending on the access size.

Byte (1 byte)	unsigned char []
Word (2 bytes)	unsigned short []
Long (4 bytes)	unsigned long []

[Remarks]

In this access, the reading of data is not posted.

F.12 SSDI_Write_Memory (Write to Memory)

[Format]

int SSDI_Write_Memory (unsigned long addr, int size, int length, void *data)

[Argument]

unsigned long	addr;	// Access address
int	size;	// Access size (1:Byte / 2:Word / 4:LONG)
int	length;	// Write data count
void	*data;	// Data storage area

[Return value]

Count of data completely written.

If the return value is different from the write data count (length), an error occurs.

[Explanation]

Writes data to memory

The type of the data storage area (data) differs depending on the access size.

Byte (1 byte)	unsigned char []
Word (2 bytes)	unsigned short []
Long (4 bytes)	unsigned long []

[Remarks]

In this access, the writing of data is not posted.

F.13 SSDI_Read_Register (Read from Register)

[Format]

```
int SSDI_Read_Register (int reg_no, unsigned long *data)
```

[Argument]

int	reg_no;	// Register number
unsigned long	*data;	// Data storage area

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Reads data from registers

[Remarks]

The register number is defined in the include file (SSDI_REGISTER.H).

"SSDI_REGISTER.H" is installed "X:YYY\LIB\896\SSDI" folder.

X:YYY It is the drive & folder which it installed SOFTUNE in

F.14 SSDI_Write_Register (Write to Register)

[Format]

```
int SSDI_Write_Register (int reg_no, unsigned long data)
```

[Argument]

int	reg_no;	// Register number
unsigned long	data;	// Data storage area

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Writes data to registers

[Remarks]

The register number is defined in the include file (SSDI_REGISTER.H).

"SSDI_REGISTER.H" is installed "X:YYY\LIB\896\SSDI" folder.

X:YYY It is the drive & folder which it installed SOFTUNE in

F.15 SSDI_Set_Interrupt (Set Interrupt Source)

[Format]

```
int SSDI_Set_Interrupt (int int_no, int sw)
```

[Argument]

int	int_no;	// Interrupt number
int	sw;	// Interrupt state (=0:OFF / =1:ON)

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Sets the interrupt state.

If an interrupt is accepted, the call back (SSDI_Interrupt_Event) is called.

[Remarks]

If the interrupt state is set ON, an interrupt request always occurs. Set the interrupt state OFF in the timing in which the interrupt source flag is cleared.

F.16 SSDI_Set_Timer (Set Timer)

[Format]

int SSDI_Set_Timer (int no, unsigned long cycle, int sw, int id)

[Argument]

int	no;	// Timer setting number
unsigned long	cycle;	// Cycle count
int	sw;	// Condition (0: Repeat/1: Only once)
int	id;	// Always set 0

[Return value]

Timer setting numbers (0 to 31)

= -1 Error

[Explanation]

Generate a timer event after an elapse of the set cycle count after the timer is set.

There are two timer setting conditions, repeat and only once.

To clear the timer setting, set the cycle count (cycle) to - 1.

If the timer setting number is set to -1, it is set to an unassigned number.

[Remarks]

F.17 SSDI_Request_Abort (Request Abort)

[Format]

```
void SSDI_Request_Abort (char *message)
```

[Argument]

```
char                *message;    // Abort message
```

[Return value]

None

[Explanation]

Requests the ISS to abort.

When an abort is requested, the ISS aborts processing when the current instruction execution is terminated.

The abort message (message) is displayed as the abort message for the debugger.

[Remarks]

F.18 SSDI_Set_Area (Set Area)

[Format]

```
int SSDI_Set_Area (int no, unsigned long start, unsigned long size, int attribute, int id)
```

[Argument]

int	no;	// Area setting number
unsigned long	start;	// Starting address of area
unsigned long	size;	// Area size
unsigned long	attribute;	// Attribute
int	id;	// Always set 0

[Return value]

Area setting numbers (0 to 31)

= -1 Error

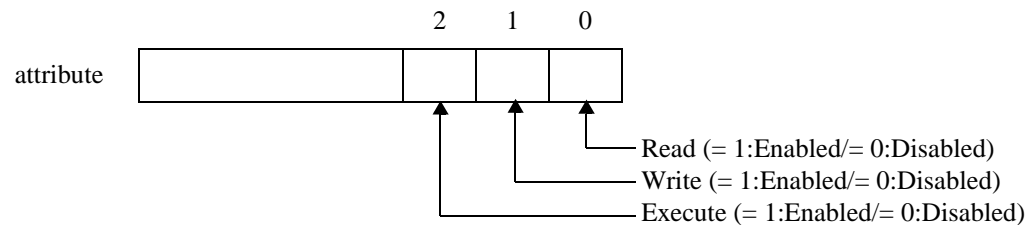
[Explanation]

Sets the area where a read event, write event or execute instruction event occurs.

An event occurs only upon access to the area in which an area is set. Set an area by using SSDI_Init (Initialize).

If the area setting number is set to -1, the area setting address/area size is valid. In other cases, set the attributes for the settings in the area setting numbers.

Set the attributes as follows:



To delete the area settings, set the number of the area to be deleted in the area setting number and the attribute to 0.

Up to 32 areas can be set. If the maximum value is exceeded, an error occurs.

[Remarks]

F.19 SSDI_Clear_Cycle (Clear cycle count)

[Format]

int SSDI_Clear_Cycle (void)

[Argument]

None

[Return value]

None

[Explanation]

Initialize total cycle number managing with ISS (Instruction Set Simulator).

[Remarks]

F.20 SSDI_Execute_MCU (Execute target program)

[Format]

void SSDI_Execute_MCU (void)

[Argument]

None

[Return value]

None

[Explanation]

Starts execution of program for ISS.

When execution of program starts it already, it is ignored.

[Remarks]

F.21 SSDI_GetCurrentExecuteAddr (Get the Address in Execution)

[Format]

unsigned long SSDI_GetCurrentExecuteAddr (void)

[Argument]

None

[Return value]

The address of currently executed instruction

[Explanation]

Get the address of the instruction currently in execution.

[Remarks]

CM25-00322-2E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL
F²MC-8L/8FX FAMILY
SOFTUNE™ Workbench
OPERATION MANUAL

June 2007 the second edition

Published **FUJITSU LIMITED** Electronic Devices
Edited Business Promotion Dept.
